XML ÉS FÉLIG-STRUKTURÁLT ADATBÁZISOK FÜGGŐSÉGEI

Doktori értekezés

Készítette:
Szabó Gyula István
okleveles matematikus

Témavezető:
Dr. Benczúr András egyetemi tanár
Az MTA doktora

INFORMÁCIÓS RENDSZEREK TANSZÉK
EÖTVÖS LORÁND TUDOMÁNYEGYETEM, INFORMATIKAI KAR

INFORMATIKA DOKTORI ISKOLA
A Doktori Iskola vezetője:
Dr. Benczúr András egyetemi tanár
Az MTA doktora

INFORMÁCIÓS RENDSZEREK DOKTORI PROGRAM
Vezető:
Dr. Benczúr András egyetemi tanár
Az MTA doktora

Budapest, 2013.

Köszönetnyilvánítás

Köszönöm témavezetőmnek, Benczúr Andrásnak, hogy tanítványául fogadott és segített számítógép programozóból informatikussá válnom. Köszönöm Demetrovics Jánosnak, hogy az Informatika Doktori Iskolába irányított. Köszönöm Kiss Attila Elemérnek, hogy bevont az Információs Rendszerek Tanszék kutatómunkájába és segített megérteni, hogy a programok hosszánál fontosabb az algoritmusuk minősége. Az Informatika Doktori Iskola nyújtotta képzés nélkül nem tudtam volna ezt a dolgozatot elkészíteni, ezért is köszönetet mondok. Köszönöm feleségemnek, Vámos Zsuzsanna okleveles alkalmazott matematikusnak, hogy mellettem állt és bíztatott amikor erre volt szükségem, és segített újra birtokba venni az informatika matematikai alapjait.

Tartalomjegyzék

1. Bevezetés

Az Internet (World Wide Web) mindennapjaink részévé vált, ide fordulunk információkért, ide helyezzük az információkat, amelyeket a világgal közölni kívánunk. Ennek a szédületes információfolyamnak ki nem mondottan standard formátumává lett az XML. Az XML eredetileg egyedi dokumentumok készítésére szolgáló eszköz volna, a gyakorlatban, mint az on-line adatközlés egyeduralkodója, adatbázis kezelő eszközzé fejlődött. A relációs adatbázisokban tárolt adattömegek Internetes közzétételéhez is az XML formátumot használják: az adatbázisok on-line elérése XML nézeteken (view) keresztül valósul meg [38]. Arra is van igény, hogy XML dokumentumok adatait relációs adatbázisokban tárolják [5, 14].
A relációs adatbázis merev sémaszerkezete nehezen illeszkedik az on-line alkalmazásokban megszokott vizuális felületek követelményeihez. Az on-line alkalmazások elvárásaihoz kapcsolódva alakultak ki az önleíró adatszerkezetek (HTML, félig-strukturált adatbázisok, SGML, XML). A relációs adatbázisban tárolt adatok megjelenítéséhez (view) adatbázis konverzió szükséges, ezen konverzióknál felmerült a relációs függőségek „konverziójaként" keletkező függőségek, azaz a félig-strukturált adatbázisok adatfüggőségei vizsgálatának igénye.
Az információs rendszerekben kezelt adatok közötti függőségek fontossága a relációs adatbázisok elméletében vált világossá. A függőségek felismerése és meghatározása segíti a „jó" séma tervezését, a feltöltött adatbázisban biztosítja az adatok integritásának megőrzését (épségi megszorítások).

Ebben a dolgozatban az XML adatbázisokon értelmezett függőségekkel (épségi megszorításokkal) foglalkozunk. Az épségi megszorítások megőrzése adatbázis transzformációk (relációs $\leftrightarrow$ XML, XML $\leftrightarrow$ XML) során hasznos módszer annak ellenőrzésére, hogy a transzformált adatbázis adattartalma az eredetiével megegyező marad-e [16, 39]. Amikor relációs adatbázisokhoz XML nézeteken keresztül férünk hozzá, különösen adatmódosítások esetében, az adatbázis konzisztenciáját az épségi megszorítások segítenek megtartani. A függőségek az XML dokumentumok tervezése során hozzájárulnak a konzisztens adatmodell kialakításához [6].
A relációs modell két általánosítását, a komplex értékű adatbázisokat valamint az önleíró adatszerkezeteket vizsgálva megállapíthatjuk, hogy az XML mindkét kritériumnak megfelel; nyilvánvalóan önleíró szerkezetű, valamint felfogható komplex értékű adatok (elemek) gyűjteményének is, ahol egy elem nyitó *jelölő*-je és záró *jelölő*-je közötti teljes jelsorozat az elem összetett értéke.
Mint félig-strukturált adatbázis, az XML egyik fő jellemzője a séma nél-

küliség, de a gyakorlat sémát követelt, így megjelennek az XML sémaleíró nyelvei, mint pl. DTD és XML-séma, beépített függőség definíciókkal (XML kulcs, idegen kulcs). A szakirodalomban számos a sémanyelveket kiegészítő függőségfogalom született meg. Az *XML kulcs* leírható az XML sémanyelvekben is, de további, a sémanyelveken túlmutató definíciókat is megfogalmaztak [29, 11]. A relációs modellből ismert tartalmazási függőségnek is vannak XML változatai [19]. Az XML numerikus megszorítást [30] és a többértékű függőséget [52] is vizsgálták. Ismereteink szerint az XML összekapcsolási függőséget még nem fogalmazták meg. A relációs modellhez hasonlóan, az XML világban is a funkcionális függőség a legnépszerűbb a kutatók között.

A funkcionális függőség (functional dependency - FD) valószínűleg a legfontosabb épségi megszorítás bármely adatmodell esetén (ahol egyáltalán értelmezhető, azaz, szinte minden modellben), de mindenképpen a legalaposabban elemzett függőség. A relációs FD definíciója természetesen adódik: egy Y attribútumhalmazon felvett értékek egy másik, X attribútumhalmazon felvett értékektől függenek, azaz, „Y az X függvénye". Másképpen fogalmazva, ha egy reláció két sora megegyezik az X-beli attribútumokon, akkor meg kell, hogy egyezzen az Y-beli attribútumokon is. A relációs adatmodellben az FD-t sokoldalúan elemezték és a normalizációs elméletben széleskörűen hasznosították [1]. Az XML funkcionális függőséget (XFD) számos különböző módon definiálták, ám nem született általánosan elfogadott meghatározás. A fő probléma a funkcionális függőség XML környezetben való definiálásakor a „sor" fogalmának hiánya. A relációs séma egy előfordulása azonos szerkezetű sorok (véges) halmaza, így természetesen adódik sor-párok kiválasztása a funkcionális függőség ellenőrzéséhez. Egy XML dokumentum szerkezetileg gyökeres fa, amelyben a belső csomópontok XML elemek (jelölők), a levelek az elemek értékei, ezért az XML világban nincs magától értetődő, természetesen adódó, általánosan elfogadott „sor" fogalom, és ha ki is választjuk az elemek valamely együttesét és „sor"-nak nyilvánítjuk őket, általában nehéz találni egy „jó" összehasonlító eljárást. Arenas és Libkin alapvető munkájukban „tree tuple" formájában (nem magyarítjuk ezt az általánosan használt XML „sor" fogalmat, részben furcsa magyar hangzása - fasor - miatt sem) találták meg az XML „sor"-okat, DTD sémaleírásra építve [6]. Vincent és munkatársai [53, 54] olyan további eseteket vizsgáltak, amelyeket a „tree tuple" modell nem kezel, ezen esetek leírására bevezették a „closest node" fogalmát (ezt sem magyarítjuk). A funkcionális függőséget séma nélküli XML fákon értelmezték (DTD-ket csak ahhoz használtak, hogy bebizonyítsák modelljük és a „tree tuple" modell ekvivalenciáját bizonyos DTD osztályok esetén). Hartmann és kollégái [25, 32] fa homomorfizmus segítségével definiálnak funkcionális függőséget, XML séma gráfot használva a függőség leírásához.

Wang (2005-ben) számos különböző XML funkcionális függőség definíciót hasonlított össze és javasolt egy új XFD koncepciót, amely egységesítené és általánosítaná az elemzett XFD-ket: mind az elemzett, mind a javasolt XFD definíciók DTD-ből vagy XML-séma leírásból képezett ösvénykifejezésekből épülnek fel [55]. Számos további XFD definíció született. Legutóbb, Hartmann és kollégái [31] szemlézték a közelmúlt XML funkcionális függőségeit, hogy az Armstrong fák koncepciójának alkalmazhatóságát ellenőrizzék az XFD-k egy osztályán.

Az XML funkcionális függőség (XFD) korai definícióit (*tree tuple*, *closest node*,...) követően számos javaslat született, az XFD továbbra is kutatási témaként szerepel. Az XFD logikai implikációja nehezen eldönthető, véges axiomatizálása az általános esetben (összetett diszjunktív reguláris kifejezések kezelésekor) nem lehetséges [6]. Valamennyi XML funkcionális függőség definíció meglehetősen bonyolult (nincs általánosan elfogadott „sor"-koncepció), összehasonlítva a „természetes" relációs funkcionális függőséggel. Az XML adatmodell esetében az FD definíciókat többnyire útkifejezésekre építik.
Látható, hogy az alapvető különbség a relációs FD és a különböző XFD fogalmak között a függőséget alkotó komponensek egymás közötti viszonyának különbözőségéből adódik: a relációs esetben a kapcsolat „horizontális", az XML esetben „vertikális" komponensek is szerepel(het)nek. Ez a különbség nem csak a funkcionális függőségekre, hanem a többi megszorításokra (tartalmazási, többértékű, összekapcsolási, ...) is érvényes, így egy, a relációsnál általánosabb sorfogalommal az XML függőségekhez hasonló, további függőségeket, épségi megszorításokat is megfogalmazhatunk. Az XML dokumentum felfogható tipizált elemek („scope") gyűjteményeként is. A függőséget egy adott típusú elem közvetlen beépülőin értelmezve jól kezelhető függőségfogalomhoz jutunk, amely egyrészt túllép a relációs modellen (ismétlődő attribútumokat, összetett és hiányos adatokat is kezelni kell), másrészt, bár alkalmas az XML adatbázis adatfüggőségeinek (egyszerűbb) megjelenítésére, de az XML szokásos, fa-szerkezettel való modellezését nem követeli meg.

Mivel az XML elemeket a sémanyelvek reguláris kifejezésekkel írják le, kézenfekvő, hogy a függőségeket egy reguláris nyelven (a nyelv mondatain) értelmezzük. Ezzel a megközelítéssel nyilván az XFD-nél (vagy más, XML függőségnél) lényegesen szűkebb fogalomhoz jutunk, hiszen így az XML hierarchikusan tagolt fa-szerkezete helyett egyetlen elem ugyan összetett, de zárt világára korlátozódunk. Mindenesetre, ezen az úton az egyes függőségek egyszerű, de elég általános definícióit sikerül megadni, amely definíciók az adatmodellek egy tág osztályán érvényesek: egyetlen feltétel van, az „adatsorok" legyenek egy reguláris nyelv mondatai, azaz, generálja azokat egy regu-

láris grammatika vagy ezzel ekvivalens módon, feleljenek meg egy reguláris kifejezésnek.

A reguláris nyelvet adatbázisként interpretáló modell számos ismert függőségtípust (funkcionális, tartalmazási, összekapcsolási) megalapozhat, az Értekezés azonban kizárólag a reguláris nyelveken értelmezhető funkcionális függőséggel (RFD) foglalkozik. Ez a függőségfogalom lényegében nem csupán egy új tagja a tág XML függőségek (funkcionális függőség esetében az XFD „XML FD") családnak. Értelmezése általános reguláris nyelveken történik, de alkalmazható az XML-re is, amennyiben az XML sémaleírás (vagy annak releváns része) kikötéseinknek megfelel (nincs több szint). Ezen nézőpont szerint, egy XML dokumentum szövegdarabok halmaza, mindegyik szövegdarab szimbólumsorozatként áll elő (a szimbólumok a szöveges adatok, adatértékek; egy adatértéket egy szimbólum reprezentál), és ezen szimbólumsorozatok típusai is egy reguláris nyelv (a duális nyelv) mondatai. (Például, egy relációs sémán értelmezett funkcionális függőség definícióját természetes módon vihetjük át egy reguláris nyelvhez rendelt duális nyelvre: a duális nyelv mondatai játsszák a séma szerepét). Ebben a megközelítésben, ha egy XML dokumentumról van szó, akkor egy adott elemtípus előfordulásait tekintjük, amelyek az XML dokumentumban szétszórtan vannak jelen. Ez a megközelítés lényegesen különbözik a (formálisan sokban hasonló) reguláris fa modelltől [36, 42], amely reguláris fa nyelvből építkezik, mert a reguláris fa nyelvek fákat kezelnek, míg a reguláris nyelvek szimbólumsorozatokat (bár a két grammatika formális leírása hasonló).
A reguláris ösvénykifejezések fontos szerepet játszanak az XML kulcs témában [29, 11]. A jelen dolgozatban tárgyalt modell, a reguláris nyelvekre építő koncepció lényegesen más mint a reguláris ösvénykifejezések modellje: az előbbi testvér-elemek tartalmát hasonlítja össze, az utóbbi szülő-gyermek viszonyokat vizsgál.

Nem szokványos jelölések

Jelsorozatok. A dolgozatban formális nyelvi fogalmakat használunk, a megszokottól némileg eltérő módon. A példákban felhasznált ábécék szimbólumai alakilag nehezen különíthetők el a környező leíró szövegtől, ezért alkalmanként, különösen a példákban, a jelsorozatok megadásakor kezdő és befejező határoló jeleket - $\lfloor, \rceil$ - használunk. Így pl. az a, b szimbólumokból képezett *abaaabbbabab* jelsorozatot $\lfloor abaaabbbabab \rceil$ formában is megadhatjuk, ahol a két határoló jel nem tartozik a jelsorozathoz, a $\lfloor, \rceil$ határoló jelek nem tartoznak a tárgyalt nyelvek ábécéihez. Ebben az esetben azonban *abaaabbbabab* nem zavaró, miután ez az a forma, amelyet a formális nyelvekkel

foglalkozó szakirodalom használ: egy tipikus, kétbetűs (a, b) ábécé használatánál világosan elkülönülnek a *betűk*, a vizsgálat tárgya ismétlődésük és a mondaton belüli helyzetük.

A dolgozatban a *szimbólum* szerepét összetett kifejezések is játszhatják, olyan összetett kifejezések, amelyek maguk is valamely „háttér" ábécé betűivel írhatók le, de a szimbólumként használt összetett kifejezések szerkezetét nem vizsgáljuk, adottnak vesszük, egy-egy összetett érték csak mint a szóbanforgó ábécé betűje jelenik meg. Ha például az ábécé „betűi" nevek és számok (például a $\{Kiss, Nagy, Török$,123,234,456$\}$ halmazból), akkor a $KissNagyTörök$ „jelsorozat" (amely három „jel"-ből áll) még felismerhető, de a szintén három jelből álló 123234456 (összetett) jelek sorozata már nehezen olvasható. Ilyenkor inkább a $\lfloor Kiss\ Nagy\ Török \rceil$, illetve a $\lfloor 123\ 234\ 456 \rceil$ formát használjuk.
Előfordul, hogy még ez a forma sem eléggé választja el az egyes szimbólumokat, például ha az ábécé a $\{$10,111,112, $Elek, Mara\}$ szimbólumokból áll, akkor a nyelv egy t mondatának szemléltetésére a $t = \lfloor 10\ 111\ Elek\ 112\ Mara \rceil$ alak helyett inkább a $t = \lfloor 10 \mid 111 \mid Elek \mid 112 \mid Mara \rceil$ formát használjuk, azaz, a „|" elválasztó jelek sem tartoznak az ábécéhez, csak a megjelenítés olvashatóságát segítik.

Több esetben használunk a nyelv ábécéjéhez tartozó kezdő és záró elhatároló jeleket (pl. $\langle, \rangle$), ilyenkor a $\langle abaaabbbabab \rangle$ jelsorozat a $\lfloor \langle abaaabbbabab \rangle \rceil$ formában is megjelenhet.

Helyettesítések. A formális nyelvekkel foglalkozó szakirodalomban az „Y szimbólumcsoport helyettesíti az X szimbólumcsoportot" szabályt az $X \to Y$ kifejezéssel jelölik, az $X \Rightarrow Y$ kifejezés azt jelenti, hogy az Y az X-ből helyettesítések sorozataként nyerhető. Ebben a dolgozatban az $X \to Y$ kifejezéssel azt fogjuk jelölni, hogy az Y funkcionálisan függ az X-től (bármit is jelentsen ez), ezért az „Y szimbólumcsoport helyettesíti az X szimbólumcsoportot" helyettesítési szabály leírására az $X \Rightarrow Y$ kifejezést fogjuk használni.

Jelsorozatok szimbólumhalmaza. A megszokottól kissé eltérő módon többször szükségünk lesz arra, hogy egy jelsorozat szimbólumaira halmazként hivatkozhassunk, például akkor, ha azt akarjuk egyszerűen kifejezni, hogy az α szimbólum „ott van" az ω jelsorozat szimbólumai között (az $\alpha \in \omega$ nem teljesen szabatos, hiszen ω nem egyszerűen szimbólumainak halmaza, a sorrend és ismétlődés is számít, nyilván $aa \neq aaa$). Bevezetjük az $[\omega]$ jelölést, az ω különböző szimbólumainak halmazára, azaz ha Σ egy nem üres ábécé és $\omega \in \in \Sigma^*$ akkor $[\omega] = \{\alpha | \alpha \in \Sigma; \exists \beta, \gamma \in \Sigma^*; \omega = \beta\alpha\gamma\}$, így, ha $[\omega] = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ akkor $\alpha_i \neq \alpha_j; 1 \leq i < j \leq n$. Nyilván $[a] = [aa] = [aaa] = \{a\}$.

Reguláris nyelv mint adatbázis

Egy relációs adatbázis sorait tekinthetjük egy reguláris nyelv mondatainak ha a reláció sorainak keletkezését figyeljük, amikor attribútumokhoz értékeket rendelünk: vesszük az attribútumok alkotta (a relációs táblázat oszlopait meghatározó) sort (például A, B, C, D, E) és az attribútumokhoz rendeléssel létrehozzuk az értékek (adatsorok) sorait (egy sorra példa a, b, c, d, e). Ez a folyamat azt jelenti, hogy az attribútumokat ismételten értékekkel (adatokkal) helyettesítjük. Azaz, legyen *R(A,B,C,D,E)* egy relációs séma, *A,B,C,D,E* az attribútumok halmaza. Akkor a $P: A \Rightarrow aB, B \Rightarrow bC, C \Rightarrow cD, D \Rightarrow dE, E \Rightarrow e$ helyettesítési szabályok (egy megfelelő reguláris grammatikából) az R adatbázis sorait generálják (1. ábra). Ez a grammatika reguláris nyelvet generál, tehát az R reláció sorai egy reguláris nyelv mondatai.
Ez a megfeleltetés a reguláris nyelv és a relációs adatbázis között nem pontos: a relációs adatbázis attribútumaihoz a potenciális értékek (attribútumonként akár különböző) tartományából bármely (a tartomány méretétől függően akár végtelen sok adatból választott) értéket rendelhetünk, így a relációnak sok, esetleg végtelen számú sora is lehet. Ezzel szemben a P helyettesítési szabályok szigorúan véve csak egy reguláris mondatot - *abcde* - állítanak elő. A formális nyelvek elméletében a tankönyvi példák leírnak végtelen számosságú reguláris nyelveket, de ezeket a nyelveket véges számú szabály generálja, a végtelen számosságot a véges elemek ismétlődése okozza. Például a P_1 : : $A \Rightarrow aB, B \Rightarrow bA, B \Rightarrow c$ helyettesítési szabályok (legyen A a kezdőszimbólum) az $ac, abac, ababac, \ldots$ mondatokat generálják. A mondatok száma nem korlátos: a véges számú helyettesítési szabály és a véges sok betű mellett ez azért lehet, mert a generált mondatok hossza nem korlátos.
Ha a reguláris nyelvekkel adatbázisokat szeretnénk modellezni akkor a helyettesítési szabályok számát nem korlátozhatjuk: a fenti P szabályhalmaz $X \Rightarrow xY$ helyettesítési szabálya az $X \Rightarrow x_1Y, X \Rightarrow x_2Y, X \Rightarrow x_3Y, \ldots$ sorozatot reprezentálja, azaz a generált nyelv mondatai $a_ib_jc_kd_le_m$ alakúak lesznek, ahol $a_i \in \{a_1, a_2, a_3, \ldots\}$, $b_j \in \{b_1, b_2, b_3, \ldots\}$ és így tovább. Az említett szabálysorozatokat úgy is megkaphatjuk, ha az $X \Rightarrow xY$ helyettesítési szabályt szabálysémaként értelmezzük. A szabálysémák komolyabb szerepet akkor kapnak, ha a helyettesítési szabályok (a nemterminális szimbólumok) száma végtelen, ezzel az esettel a 10.5. Fejezetben foglalkozunk.

Egy féligstrukturált adatsor <attribútum: érték> párok rendezett listája. Ezt az adatsort egy reguláris nyelv mondataként is értelmezhetjük, az értékek a terminális szimbólumok és az attribútumnevek a nemterminális szimbólumok. Az attribútumnevek sorozatának megfelelnek a helyettesítési szabályok bal oldalain álló nemterminális szimbólumok amelyeket a sorban következő terminális szimbólum (a relációs sor következő adatértéke) elfoga-

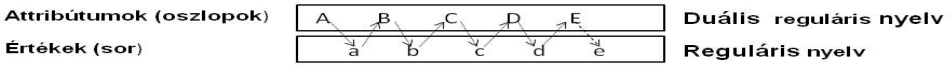
dásához használtunk. Az adatok listája a reguláris mondat, az attribútumok listája a duális mondat a 2. ábrán láthatóan. Ezzel a megközelítéssel egy XML elemtípus előfordulásait (esetleg különböző sémájú) adatsorok (véges) halmazaként foghatjuk fel. Ezen sémák mindegyike megfelel az elemtípus definíciójában adott reguláris kifejezésnek.

A relációs sor sémája szerkezetileg attribútumlista. Egy kiterjesztett (általánosított) sor egy reguláris nyelv egy mondata, ennek a struktúrája (séma) a társított (ugyancsak reguláris) duális nyelv egy mondata. A kiterjesztett reláció kiterjesztett sorokból áll; minden sort a saját sémája ír le, annak felel meg. A sémának megfelelő kiterjesztett sorok halmazát a kiterjesztett reláció egy előfordulásának nevezzük. A kiterjesztett sorokat vagy egy reguláris grammatika generálja vagy egy reguláris kifejezés szerint állnak elő.
Egy függőséget mint az adatok egy halmazán értelmezett megszorítást definiálunk. Egy relációs adatbázis relációkból, névvel ellátott adatgyűjteményekből áll. Egy relációban az adattípus egy rögzített sortípus. A függőség szintaktikai definícióját az adatstruktúrák szintaktikus elemeiből konstruáljuk. A megszorítás szemantikáját az adatgyűjtemény egy előfordulásán értelmezzük.

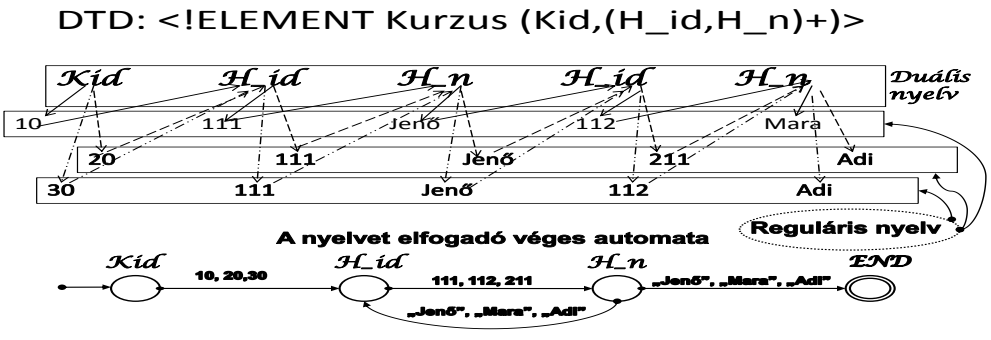
Mivel egy XML dokumentum adattípusait elem szinten definiáljuk, a legtermészetesebb adatgyűjtemény egy XML dokumentumban az azonos DTD vagy XML-séma elemdeklarációhoz tartozó elemek halmaza. Ez a halmaz egy kiterjesztett reláció soraiból áll.

Eredmények. A dolgozatban ismertetett eredmények a reguláris nyelveken értelmezett kiterjesztett relációk újonnan, a [48] cikkben és a cikkhez kapcsolódó FOIKS 2012 konferenciaelőadásban bevezetett fogalmához kapcsolódnak. Funkcionális függőséget definiálunk a kiterjesztett relációk adatbázisán, kimutatjuk, hogy ezek az RFD-nek (reguláris FD) nevezett függőségek értelmezhetők az XML dokumentumokon, algoritmust adunk meg logikai implikációjuk eldöntésére és az eldöntő algoritmust felhasználva megadjuk a logikai implikáció Armstrong-típusú axiomatizációját. Példákon hasonlítjuk össze az RFD-t a legfontosabb XFD koncepciókkal, megmutatjuk, hogy az XFD-től eltérően a véges axiomatizáció az RFD logikai implikációjára tetszőleges számú diszjunkciót tartalmazó reguláris kifejezések esetén is lehetséges. Vizsgáljuk a számlálókkal bővített, halmazkonstruktorokat használó és a végtelen ábécét kezelő RFD esetét is. Az RFD fogalmát átvisszük környezetfüggetlen nyelveken alapuló adatbázisokra.

Az Értekezés felépítése. A 2. Fejezet informálisan bevezeti a kiterjesztett relációkat, a 3. Fejezet a reguláris nyelveket felismerő automaták konstrukcióját tárgyalja, reguláris grammatika vagy reguláris kifejezés alap-



1. ábra. Relációs DB mint reguláris nyelv



2. ábra. XML elemek mint egy reguláris nyelv mondatai

ján. A 4. Fejezet a duális nyelvre és a kiterjesztett relációra, azok kiterjesztett attribútumaira ad formális definíciót példákon illusztrálva. Az 5. Fejezet a kiterjesztett relációkon elvégezhető projekciókhoz kapcsolódó összehasonlító eljárásokat ad meg. A 6. Fejezet a reguláris funkcionális függőséget ismerteti és elemzi, logikai implikációját és axiomatizációját oldja meg. A 7. Fejezet az RFD-t XML sémanyelvekre alkalmazza. A 2-7. Fejezetek főbb eredményeit publikáltam, konferenciaelőadásokon [48, 50] kívül a dolgozat készítésekor megjelenés alatt van egy hosszabb folyóiratcikk a 2-7. Fejezetek anyagából. A 10. Fejezet az RFD további érdekes tulajdonságait vizsgálja (bizonyos határok, megkötések feloldása, végtelen számosságú ábécék kezelése), eddig nem publikált eredményeket ismertet. A 11. Fejezet a reguláris funkcionális függőség környezetfüggetlen nyelvekre alkalmazásának konferenciaelőadáson már ismertetett koncepcióját tárgyalja [49].

2. Kiterjesztett relációk reguláris nyelveken

Ebben a fejezetben informálisan, példák segítségével bevezetjük a reguláris nyelveken értelmezett kiterjesztett relációkat, a formális definíciót később, a 4. Fejezetben adjuk meg.

A függőségek meghatározásához reguláris nyelveken, mint adatbázisokon, szükségünk lesz néhány fogalomra. Legyen $G = (N, T, S, P)$ reguláris grammatika, ahol N a nemterminális szimbólumok, T a terminális szimbólumok és P a helyettesítési szabályok diszjunkt, véges halmazai. Minden szabály P-ben $X \Rightarrow xY; X, Y \in N; x \in T$ alakú. (Itt és a továbbiakban a „$\Rightarrow$” jelölésst használjuk a szimbólum-helyettesítés jelzésére a helyettesítési szabályokban, hogy elkerüljük a funkcionális függőséget jelölő „$\rightarrow$” kétféle felhasználásából adódó zavart). Az $X \Rightarrow xY$ szabályt felfoghatjuk szabálysémaként is, amennyiben x egy tetszőleges terminális szimbólumot reprezentál abból az értéktartományból, amely az X nemterminális szimbólumhoz tartozik. Az $X \Rightarrow xY$ helyettesítési szabályséma az $X \Rightarrow x_1Y, X \Rightarrow x_2Y, X \Rightarrow x_3Y, \ldots$ sorozatot képviseli ahol az $x_1, x_2, x_3, \ldots$ értékek az X nemterminális szimbólumot helyettesítő terminális szimbólumok.
Jelölje $L(G)$ a generált reguláris nyelvet, $M(G)$ az $L(G)$-et elfogadó véges automatát (FSA - finite state automaton). Definiáljuk az $L(G)$ nyelv duális nyelvét a következőképpen: legyen $\mathrm{w} \in L(G)$ egy mondat amelynek generálásakor a $S \Rightarrow d_0 N_1, N_1 \Rightarrow d_1 N_2, \ldots, N_k \Rightarrow d_k$ szimbólumhelyettesítések hajtódtak végre (rövidítve, $Sd_0N_1d_1N_2 \ldots d_{k-1}N_kd_k$). A $\mathrm{w} = d_0d_1 \ldots d_{k-1}d_k$ mondat generálásakor hivatkozott nemterminális szimbólumok sorozata egy másik mondatot, a $W = S\,N_1N_2 \ldots N_k$-t hozza létre.

A helyettesítésekben szereplő nemterminális szimbólumokból felépülő W mondatot a w duális (társított) mondatának nevezzük. A w mondatot kiterjesztett sornak nevezzük, és W a w kiterjesztett sor típusa (sor-típus konstruktora). A w kiterjesztett sor típusa $< S : D_0, N_1 : D_1, \ldots, N_k : D_k >$, ahol D_i az N_i nemterminális szimbólumhoz rendelt értéktartomány, a relációs adatmodellhez hasonlóan. A kiterjesztett sorok halmaza a kiterjesztett reláció. Az eredeti $L(G)$ reguláris nyelv több mondata is tartozhat egy rögzített duális mondathoz. Ezen mondatok - mint kiterjesztett sorok - alkotják a kiterjesztett reláció egy előfordulását.

2.1 Megjegyzés
Az $L(G)$ reguláris nyelv mint kiterjesztett reláció előfordulásait egy-egy duális mondathoz tartozó reguláris mondatokra (azok egy részhalmazára) korlátoztuk, erre azért van szükség, hogy a kiterjesztett reláción értelmezett függőségek ellenőrzésénél azonos szerkezetű sorokkal dolgozhassunk és így elkerülhessük, többek között, a hiányos (null) adatokon való függőségvizsgálatot. Az előfordu-

lás fogalmának ezt a leszűkítését a 10.2. Fejezetben M2 megkötésnek nevezzük (és elemezzük feloldásának következményeit). Ez a megkötés véges számú terminális szimbólum (= véges sok helyettesítési szabály; a nemterminálisok száma mindig véges) esetén az előfordulás sorainak számát is végessé korlátozza (ha I a W duális mondathoz tartozó előfordulás, akkor $|I| \leq |T|^{|W|}$), ezért a szokásos kikötés (az előfordulás a sorok valamely véges halmaza) elhagyható. Végtelen terminális ábécék esetével a 10.5. Fejezetben foglalkozunk.

Az $L(G)$ nyelv mondataihoz társított duális mondatokból épül fel az $L(G)$ nyelv duális nyelve, amelyet $D(L)$-lel jelölünk. Egy reguláris nyelv duális nyelve is reguláris; grammatikája, $G' = (N', N, S', P')$ az eredeti nyelv grammatikájából, G-ből könnyen meghatározható (nyilvánvalóan, $D(L) = = L(G'))$.

A $D(L)$ duális nyelvet az $L(G)$ nyelvet elfogadó $M(G)$ véges állapotú automatával reprezentálhatjuk. Az $M(G)$ automata gráfjának minden S-ből induló és egy elfogadó állapotban végződő bejárása egy duális mondatot ad meg. Az $L(G)$ nyelvet elfogadó automata gráf reprezentációját is $M(G)$-vel jelöljük.

Az $M(G)$ gráf mindegyik csúcsa az N (szimbólum)halmaz egy szimbólumának felel meg. Van egy speciális csúcs, $\mathcal{START}$, amely a kezdőállapotot jelöli, és egy másik, speciális csúcs, $\mathcal{END}$ amely a befejező (elfogadó) állapotot képviseli. Az $M(G)$ gráf élei a P halmaz helyettesítési szabályainak feleltethetők meg: az $A \Rightarrow aB$ szabály azt jelenti, hogy az A csúcsból megy egy irányított él a B csúcsba. Ez az él bármely terminális szimbólumot reprezentálhat (erre az élre az összes terminális szimbólumot ráírhatnánk) abból az értéktartományból amelyet az A nemterminális szimbólumhoz rendeltünk. Azaz, az $M(G)$ gráf egyszerű, nem a szabályokat, hanem a szabálysémákat reprezentálja. Az $A \Rightarrow x$ szabály azt jelenti, hogy az A csúcsból megy egy irányított él az $\mathcal{END}$ csúcsba. Az $M(G)$ gráf minden $\mathcal{START}$-ból induló és $\mathcal{END}$-ben végződő bejárása egy duális mondatot reprezentál.

Lássunk egy relációs példát.

2.1. Példa. *Legyen R(A,B,C,D,E) egy relációs séma, úgy, hogy*

A,B,C,D,E az attribútumok halmaza,
***dom** jelöli az attribútumok közös értéktartományát (az attribútumokhoz különböző értéktartományokat is rendelhetünk).*

Legyen továbbá G=(N,T,S,P) reguláris grammatika, ahol

$N = \{R, A, B, C, D, E, \mathcal{ENT}\}$ *a nemterminális szimbólumok halmaza,*
$T = \{\langle, \rangle\} \cup \mathbf{dom}$ *a terminális szimbólumok (értékek) halmaza,*
$S = R$ *a kezdő (start) szimbólum,*
$P =$
$= \{R \Rightarrow \langle\, A, A \Rightarrow \mathbf{a}B, B \Rightarrow \mathbf{b}C, C \Rightarrow \mathbf{c}D, D \Rightarrow \mathbf{d}E, E \Rightarrow \mathbf{e}\,\mathcal{ENT}, \mathcal{ENT} \Rightarrow\rangle\}$*;*
(ahol $\mathbf{a}, \mathbf{b} \ldots \subset_{fin} \mathbf{dom}$*) a helyettesítési szabályok (szabálysémák).*

Az $X \Rightarrow \mathbf{x}Y$ *„szabállyal" (valójában szabálysémával), ahol* $\mathbf{x} \subset_{fin} \mathbf{dom}$ *(*$\subset_{fin}$ *véges részhalmazt jelöl) a* $\{X \Rightarrow x'Y | x' \in \mathbf{x}\}$ *szabályhalmazt tömörítettük. Az L(G) nyelv, amelyet a G grammatika generál,* $\langle abcde\rangle\,; a \in \mathbf{a}, \ldots, e \in \mathbf{e}$ *sorozatokból áll, minden ilyen sorozat („sor") a* ***dom*** *értéktartományból vett értékek (szimbólumok) összefűzéseként (konkatenáció) áll elő, és az L(G) nyelv minden részhalmaza az R reláció egy előfordulása. Mivel az* $\mathbf{a}, \mathbf{b} \ldots \mathbf{e}$ *halmazok végesek,* $|L(G)| \leq |\mathbf{a}| \times |\mathbf{b}| \times \ldots \times |\mathbf{e}|$*. Nyilván az R reláció bármely előfordulását megkaphatjuk mint az* $\mathbf{a}, \mathbf{b} \ldots \mathbf{e}$ *halmazok alkalmas megválasztásával generált L(G) nyelv egy részhalmazát.*
A terminális és nemterminális jelek megfelelő választásával a gyakorlatban előforduló relációs adatbázist kapunk.
Legyen MTel(Vezetéknév,Keresztnév,Telefonszám, Emailcím,Szobaszám) egy munkahelyi telefonnévsor sémája, akkor a G=(N,T,S,P) reguláris grammatika által generált nyelv az MTel adatbázist generálja, ha

$N = \{MTel, Vezetéknév, Keresztnév, Telefonszám, Emailcím,$
$\quad Szobaszám, \mathcal{ENT}\}$
a nemterminális szimbólumok halmaza,
$T = \{\langle, \rangle\} \cup keresztnevek \cup telefonszámok \cup emailcímek \cup szobaszámok$
a terminális szimbólumok (értékek) halmaza,
(ahol $vezetéknevek = \{Kiss, Nagy, \ldots\}$
$keresztnevek = \{Péter, Ida, \ldots\}$
$telefonszámok = \{21512, 21525, \ldots\}$
$emailcímek = \{p.kiss@xyz.hu, i.nagy@xyz.hu, \ldots\}$
$szobaszámok = \{2.512, 2.525, \ldots\}$*)*
$S = MTel$ *a kezdő (start) szimbólum,*
$P = \{MTel \Rightarrow \langle\, Vezetéknév, Vezetéknév \Rightarrow vezetéknevek\, Keresztnév,$
$Keresztnév \Rightarrow keresztnevek\, Telefonszám,$
$Telefonszám \Rightarrow telefonszámok\, Emailcím,$
$Emailcím \Rightarrow emailcímek\, Szobaszám,$
$Szobaszám \Rightarrow szobaszámok\, \mathcal{ENT}, \mathcal{ENT} \Rightarrow\rangle\}$*;*

(ahol vezetéknevek, keresztnevek . . . a megfelelő terminális jel tartományok tetszőleges elemét reprezentálják) a helyettesítési szabályok (szabálysémák).

Ezen grammatika által generált reguláris nyelvhez egyetlen duális mondat kapcsolódik:

$\lfloor\langle$ **Vezetéknév Keresztnév Telefonszám Emailc*ím* Szobaszám** $\rangle\rceil$

A generált reguláris nyelv néhány mondata:

$\lfloor\langle$ *Kiss Péter* 21512 *p.kiss@xyz.h* 2.512 $\rangle\rceil$
$\lfloor\langle$ *Kiss Péter* 21512 *p.kiss@xyz.h* 2.525 $\rangle\rceil$
$\lfloor\langle$ *Kiss Péter* 21512 *p.kiss@xyz.h* . . . $\rangle\rceil$
$\lfloor\langle$ *Nagy Ida* 21525 *i.nagy@xyz.h* 2.512 $\rangle\rceil$
$\lfloor\langle$ . . . . . 21 . . . . . @*xyz.h* . . . $\rangle\rceil$

A nyelv mondatainak részhalmazai a vállalati telefonjegyzék adatbázis előfordulásainak tekinthetők (1. táblázat): a nemterminális szimbólumok megfelelnek az adatbázis attribútumainak, a helyettesítő terminális jelek pedig az attribútumok értékeinek.

Vezetéknév	**Keresztnév**	**Telefonszám**	**Emailcím**	**Szobaszám**
Kiss	Péter	21512	p.kissxyz.h	2.512
Nagy	Ida	21525	i.nagyxyz.h	2.525
. . .	. . .	21. . .	. . .xyz.h	. . .

1. táblázat. A *MTel* munkahelyi telefonlista adatbázis egy előfordulása

A következő, XML jellegű példa kissé bonyolultabb, rekurzív reguláris grammatikán alapszik. Egy reguláris nyelv grammatikáját rekurzívnak nevezzük ha tartalmaz $A \Rightarrow aB; B \Rightarrow bC; \ldots; X \Rightarrow xY; Y \Rightarrow yA$ alakú helyettesítési szabály láncot (speciális eset az $A \Rightarrow aA$ egyelemű lánc). Könnyen látható, hogy egy reguláris grammatika pontosan akkor rekurzív, ha a vele ekvivalens reguláris kifejezés is az, azaz, ha a Kleene * előfordul benne.

2.2 Megjegyzés
Itt és a továbbiakban azt mondjuk, hogy egy XML dokumentum „megfelel" egy XML sémadefiníciónak, ha a gyökérjelölők megegyeznek és a jelölők által meghatározott értékek a sémadefiníció azonos nevű elemtípusa generálta reguláris nyelv mondataiból kerülnek ki. Ezt a fogalmat abban az értelemben használjuk ahogy Arenas és Libkin a „conformance" fogalmát definiálják [6]-ben).
Az XML dokumentumok az elemeken kívül XML attribútumokat is tartalmazhatnak. A továbbiakban, hacsak külön nem jelezzük, feltételezzük, hogy a szóbajövő XML dokumentumok és sémaleírások XML attribútumokat nem használnak.

2.2. Példa. *A 3. ábra XML dokumentuma egy könyvnyilvántartó adatbázisból vett részlet. Ez az XML dokumentum az 5. ábrán megadott DTD sémának felel meg.*

Az 5. ábra Könyv ELEMENT! deklarációja, mint reguláris kifejezés a $G=$ $= (N, T, S, P)$ *reguláris grammatikát definiálja, ahol*

$N = \{Könyv, Cím, Szerző, Isbn, Tankönyv, Év, Kiadó, \mathcal{ENDS}\}$
$T = \{\langle, \rangle\} \cup címek \cup szerzők \cup isbn \cup igennem \cup évek \cup kiadók$ *a terminális szimbólumok (értékek) halmaza,*
ahol
$címek = \{$*Foundations of Databases, A First Course in Database Systems, Computational Complexity,* $\ldots\}$
$szerzők = \{S.Abiteboul, R.Hull, V.Vianu, J.D.Ullman, J.Widom, C.H.Papadimitriou, \ldots\}$
$isbn = \{0-201-53771-0, 0-13-035300-0, 978-0-201-53082-7, \ldots\}$
$igennem = \{igen, nem\}$
$évek = \{1994, 1995, 2002, \ldots\}$
$kiadók = \{$*Addison-Wesley, Prentice-Hall,* $\ldots\}$
$S = Könyv$
$P = \{Könyv \Rightarrow \langle Cím, Cím \Rightarrow címek\, Szerző,$
$Szerző \Rightarrow szerzők\, Szerző, Szerző \Rightarrow szerzők\, Isbn,$
$Isbn \Rightarrow isbn\, Tankönyv, Tankönyv \Rightarrow igennem\, Év,$
$Év \Rightarrow évek\, Kiadó, Kiadó \Rightarrow kiadók\, \mathcal{ENDS}, \mathcal{ENDS} \Rightarrow \rangle\}$
ahol $címek, szerzők \ldots$ *terminális jel tartományok, amelyek segítségével a helyettesítési szabályok mint szabálysémák adhatók meg.*

Legyen $G' = (N', T', S', P')$ *a* G*-hez társított duális grammatika az előzőekben kifejtett duális nyelv fogalom szerint, ahol*

```
<Könyvek>
<Könyv>
<Cím>Foundations of Databases</Cím>
<Szerző>S. Abiteboul</Szerző>
<Szerző>R. Hull</Szerző>
<Szerző>V. Vianu</Szerző>
<Isbn>0-201-53771-0</Isbn>
<Tankönyv>nem</Tankönyv>
<Év>1995</Év>
<Kiadó>Addison-Wesley</Kiadó>
</Könyv>
<Könyv>
<Cím>A First Course in Database Systems</Cím>
<Szerző>J. D. Ullman</Szerző>
<Szerző>J. Widom</Szerző>
<Isbn>0-13-035300-0</Isbn>
<Tankönyv>igen</Tankönyv>
<Év>2002</Év>
<Kiadó>Prentice-Hall</Kiadó>
</Könyv>
<Könyv>
<Cím>Computational Complexity</Cím>
<Szerző>C. H. Papadimitriou</Szerző>
<Isbn>978-0-201-53082-7</Isbn>
<Tankönyv>igen</Tankönyv>
<Év>1994</Év>
<Kiadó>Addison-Wesley</Kiadó>
</Könyv>
</Könyvek>
```

3. ábra. Egyszerű XML dokumentum (2.2. Példa)

Duális mondat:
$\lfloor\!\langle$ **Cím Szerző Szerző Szerző Isbn Tankönyv Év Kiadó** $\rangle\!\rceil$
Reguláris mondat:
$\lfloor\langle\ Foundations \ldots\ Abiteboul\ Hull\ Vianu\ 0-\ldots\ nem\ 1995\ A-W\ \rangle\rceil$
Duális mondat: $\lfloor\!\langle$ **Cím Szerző Szerző Isbn Tankönyv Év Kiadó** $\rangle\!\rceil$
Reguláris mondat: $\lfloor\langle\ First \ldots\ Ullman\ Widom\ 0-\ldots\ nem\ 2002\ P-H\ \rangle\rceil$
Duális mondat: $\lfloor\!\langle$ **Cím Szerző Isbn Tankönyv Év Kiadó** $\rangle\!\rceil$
Reguláris mondat:
$\lfloor\langle\ Computational \ldots\ Papadimitriou\ 978-\ldots\ igen\ 1994\ A-W\ \rangle\rceil$

4. ábra. A *Könyvek* adatbázis reguláris mondatokként

```
<!DOCTYPE Könyvek [
<!ELEMENT  (Cím, Szerző+, Isbn, Tankönyv, Év, Kiadó)>
<!ELEMENT Cím (#PCDATA)>
<!ELEMENT Szerző (#PCDATA)>
<!ELEMENT Isbn (#PCDATA)>
<!ELEMENT Tankönyv (#PCDATA)>
<!ELEMENT Év (#PCDATA)>
<!ELEMENT Kiadó (#PCDATA)>
]>
```

5. ábra. DTD sémadefiníció a 3. ábrán lévő XML dokumentumhoz.

$N' =$
$= \Big\{KÖNYV, CÍM, SZERZŐ, ISBN, TANKÖNYV, ÉV, KIADÓ, \mathcal{ENDE}\Big\}$
$T' = \{\langle, \rangle\} \cup \Big\{Cím, Szerző, Isbn, Tankönyv, Év, Kiadó\Big\}$
$S' = KÖNYV$
$P' = \Big\{KÖNYV \Rightarrow \Big\langle CÍM, CÍM \Rightarrow Cím\, SZERZŐ,$
$SZERZŐ \Rightarrow Szerző\, SZERZŐ, SZERZŐ \Rightarrow Szerző\, ISBN,$
$ISBN \Rightarrow Isbn\, TANKÖNYV, TANKÖNYV \Rightarrow Tankönyv\, ÉV,$
$ÉV \Rightarrow Év\, KIADÓ, KIADÓ \Rightarrow Kiadó\, \mathcal{ENDE}, \mathcal{ENDE} \Rightarrow \Big\rangle\Big\}$

A G grammatika az L(G) nyelvet generálja, amelynek mondatai XML fragmentumok (szövegdarabok), míg a társított duális nyelv, L(G'), mondatai összefűzött XML elemnevek.

Utalva a fejezet elején elmondottakra, a duális nyelv szimbólumai hasonló szerepet töltenek be, mint a relációs adatmodell attribútumai. A relációs adatbázis modellben az adatokat táblázatokban jelenítjük meg: ezekben a táblázatokban a sorok különböző értékeket hordoznak, míg az oszlopok különböző attribútumoknak felelnek meg. Minden sornak azonos az attribútumok által leírt szerkezete. Az oszlopok (mindegyik attribútumhoz pontosan egy tartozik) adatokat (értékeket) tartalmaznak egy közös, vagy attribútumonként változó értéktartományból. Ezek a táblázatok a relációk. Egy adott reláció (séma) attribútumai különbözőek és a sorrendjük közömbös.

Az itt bevezetett (reguláris) adatmodell duális mondatokat használ sémaként a kiterjesztett relációhoz; ez a sémafogalom a relációs modell sémától három lényeges jellemzőben tér el:

1. a kiterjesztett reláció sémája több mint egy sor-típust tartalmazhat,
2. a kiterjesztett sor attribútumai ismétlődhetnek,
3. az attribútumok sorrendje rögzített.

A kiterjesztett reláció és attribútumai formális definícióját később, a 4. Fejezetben adjuk meg.

3. Reguláris nyelvet elfogadó véges automata konstrukciója

Ebben a fejezetben reguláris nyelvekhez rendelünk őket felismerő véges automatákat (finite state automaton - FSA) abból a célból, hogy megalapozzuk a kiterjesztett relációk és a reguláris függőségek formális definícióját.

A reguláris nyelveken ható reguláris függőség definícióját a reguláris nyelvhez társított duális nyelvre vonatkoztatva szeretnénk megadni. Ha a reguláris nyelvet egy grammatika állítja elő akkor a duális nyelvet könnyen elő tudjuk állítani a 2. Fejezetben mondottak szerint. Azt is láttuk, hogy a duális nyelv megjeleníthető a megfelelő véges automata gráfján. A 3.1. Algoritmus egy jól ismert eljárást ad az automata konstruálására a nyelv grammatikájából:

3.1. Algoritmus. *Reguláris Grammatikából FSA.*
Input: G reguláris grammatika, amely az $L(G)$ nyelvet generálja;
Output: $M(G)$ véges automata, amely L(G)-t fogadja el;
Legyen $G = (N, T, S, P)$,
akkor a megfelelő FSA: $M(G) = (N, T, \delta, S, \mathcal{ENT})$, ahol

N az M belső állapotainak (véges) halmaza,
T a bemenő ábécé,
δ az átmenetfüggvény: minden $X, Y \in N, x \in T$-hez legyen $\delta(X, x) = Y$
akkor és csak akkor, ha van egy helyettesítési szabály $X \Rightarrow xY \in P$,
S a kezdő (induló) állapot,
$\mathcal{ENT}$ az (egyetlen) vég(elfogadó)állapot.

A 3.1. Algoritmus nemdeterminisztikus véges automatát (NFA) készít, ha vannak olyan helyettesítési szabályok, amelyek baloldalán azonos nemterminális szimbólumok, jobboldalán viszont különböző nemterminális szimbólumok állnak azonos terminális szimbólummal, azaz, például $A \Rightarrow aB, A \Rightarrow aC \in P$ valamely $A, B, C \in N; a \in T$ esetén. Minden más esetben a konstruált automata determinisztikus (DFA).

3.1 Megjegyzés
Legyen $G = (N, T, S, P)$ relációs típusú reguláris grammatika a 2.1. Példában tárgyaltak szerint. Ebben az esetben az $M(G)$ véges automata, amely a generált $L(G)$ reguláris nyelvet fogadja el, gyökeres, „lineáris" egyszerű irányított gráf amely kört nem tartalmaz. Az $X \Rightarrow xY$ szabály (szabály séma) azt

jelenti, hogy az X csúcsból, amely az X relációs attribútumot reprezentálja, egy irányított él vezet az Y csúcsba, amely éppen az Y relációs attribútumot reprezentálja, és ez az él az X relációs attribútum értéktartományából vett összes terminális szimbólumot képviseli.

A helyettesítési szabályok között szerepel az $S \Rightarrow sA$ szabály, ahol A a relációs séma első attribútumának megfelelő nemterminális szimbólum, így S a gráf gyökere.

A reguláris függőség definíciójához szükségünk van a reguláris nyelvhez társított duális nyelvre. Ha a reguláris nyelvet egy reguláris nyelvtan állítja elő, akkor a 3.1. Algoritmussal konstruálhatjuk a duális nyelvet reprezentáló véges automatát. Egy reguláris nyelvet megadhatunk a nyelv ábécéjén képezett reguláris kifejezés segítségével is, azaz, szükségünk lesz egy módszerre a duális nyelv elfogadó automatájának konstruálására a reguláris nyelvet leíró reguláris kifejezés alapján is.

A formális nyelveket tanulmányozó kutatók (formal language community) komoly erőfeszítést tettek hatékony módszerek kifejlesztésére amelyek reguláris kifejezésekből a reguláris kifejezéssel leírt reguláris nyelvet elfogadó automatákat konstruálnak [57]. Az algoritmusok két csoportba oszthatók a konstruált automata munkamódszere szerint: nemdeterminisztikus (NFA, például Glushkov automata [22]) és determinisztikus (DFA, például Brzozowski konstrukciója [10]). Mi itt Berry és Sethi klasszikus algoritmusát [8] vesszük kölcsön, amely hatékonyan konstruál egy determinisztikus automatát egy páronként különböző szimbólumokból álló reguláris kifejezésből. Nézzük meg a konstrukciót részleteiben.

3.1 Definíció (A reguláris kifejezés szintaxisa)
Legyen Ω szimbólumok véges halmaza (ábécé), akkor egy E reguláris kifejezés Ω felett (jelölésben $E \in RE_\Omega$, vagy egyszerűen $E \in RE$, ha Ω a szövegkörnyezetben egyértelmű) rekurzíven definiálható a következőképpen:
$E ::= 0|1|\alpha|E+E|E \circ E|E^*|E^?|E^+$
ahol $\alpha \in \Omega$

Ha nem okoz félreértést, a reguláris kifejezésekben az összekapcsolás (konkatenáció) $\circ$ jelét a szokásoknak megfelelően elhagyjuk.
Az E reguláris kifejezés az L(E)-vel jelölt reguláris nyelvet generálja. L(0) az üres nyelv (amely egyetlen mondatot sem tartalmaz), L(1) az a nyelv, amely az ϵ üres sorozatot tartalmazza egyedüliként. Megjegyezzük, hogy 0 és 1 nem az Ω ábécéből vett szimbólumok: 0 reprezentálja az üres reguláris kifejezést, 1 jelenti az üres sorozatot, ϵ-t.

$L(E+F) = L(E) \cup L(F)$. $L(E \circ F)$ az L(E)-ből és L(F)-ből vett mondatok összekapcsolásával létrejött mondatok halmaza. $L(E^*)$ megfelel az L(E)-ből vett nulla vagy több mondat összekapcsolásával képezett mondatok halmazának (azaz, $\epsilon \in L(E^*)$). $L\left(E^?\right)$ jelenti az L(E)-ből vett nulla vagy egy mondatból (mondatokból) alkotott halmazt (azaz, $\epsilon \in L\left(E^?\right)$). $L(E^+)$ megfelel az L(E)-ből vett egy vagy több mondat összekapcsolásából keletkezett mondatok halmazának (azaz, $\epsilon \in L(E^+)$ akkor és csak akkor, ha $\epsilon \in L(E)$).

Beery és Sethi jelöléseit [8] használva definiáljuk a $\Delta : RE \mapsto \{0,1\}$ függvényt a következők szerint:
Legyen $E \in RE$, akkor
$\Delta(E) = 1$, ha $\epsilon \in L(E)$, egyébként 0. Akkor
legyen $\alpha \in \Omega, E, F \in RE$, akkor
$\Delta(0) = 0$, $\Delta(1) = 1$, $\Delta(\alpha) = 0$,
$\Delta(E+F) = \Delta(E) + \Delta(F)$,
$\Delta(E \circ F) = \Delta(E) \circ \Delta(F)$,
$\Delta(E^*) = 1$, $\Delta\left(E^?\right) = 1$, $\Delta(E^+) = \Delta(E)$

A Berry-Sethi konstrukció a reguláris kifejezések deriváltjainak fogalmán alapszik. Az $E \in RE$ reguláris kifejezés $\alpha \in \Omega$ szerinti deriváltja (jelölve $\alpha^{-1}E$) az a reguláris kifejezés, amely az $\{\omega | \alpha\omega \in L(E)\}$ halmazt generálja. Például, ha $E = (\alpha\beta\alpha + \alpha\beta^+)$ akkor $\alpha^{-1}E = (\beta\alpha + \beta^+)$.

3.2 Definíció (Reguláris kifejezés deriváltja)
Legyen $E, F \in RE, \alpha, \beta \in \Omega$, akkor az E deriváltja α szerint, jelölve $\alpha^{-1}E$, a következők szerint áll elő:
$\alpha^{-1}1 = 0$, $\alpha^{-1}0 = 0$, $\alpha^{-1}\alpha = 1$, $\alpha^{-1}\beta = 0$, *ha* $\alpha \neq \beta$,
$\alpha^{-1}(E+F) = \alpha^{-1}E + \alpha^{-1}F$,
$\alpha^{-1}(E \circ F) = \alpha^{-1}E \circ F + \Delta(E) \circ \alpha^{-1}F$,
$\alpha^{-1}(E^*) = \alpha^{-1}E \circ E^*$,
$\alpha^{-1}\left(E^?\right) = \alpha^{-1}E$,
$\alpha^{-1}(E^+) = \alpha^{-1}E \circ E^*$.

3.3 Definíció (Reguláris kifejezés deriváltjának kiterjesztése)
Legyen $E \in RE, \alpha \in \Omega$, akkor az E reguláris kifejezés α szerinti deriváltját az E-nek az $L(E)$ mondatai szerinti deriváltjaira a következők szerint terjesztjük ki:
$\epsilon^{-1}E = E$, $(\omega\alpha)^{-1}E = \alpha^{-1}(\omega^{-1}E)$.

3.1 Propozíció (Berry-Sethi [8])
Legyen E' reguláris kifejezés amelyet E-ből szimbólumainak megkülönböztető jelölésével kapunk. Ha M' az $L(E')$-t elfogadó automata, akkor az M automata, amelyet M'-ből úgy kapunk, hogy az átmeneteket címkéző szimbólumok jelölését eltávolítjuk, egy, az $L(E)$ nyelvet elfogadó, nemdeterminisztikus automata lesz.

3.1 Tétel (Berry-Sethi [8])
Legyen $E \in RE, \alpha \in \Omega, \omega \in \Omega^*$, akkor mindegyik $(\omega\alpha)^{-1} E$ derivált vagy 0, vagy egyértelmű a + művelet szempontjából, asszociativitás, kommutativitás, és idempotens tulajdonságra tekintet nélkül.

Az előbbi tétel felhasználásával, meg tudunk konstruálni egy automatát, amelynek állapotai a reguláris kifejezés szimbólumainak felelnek meg, ha a szimbólumok páronként különbözőek. (Egy reguláris kifejezést szimbólumaiban páronként különbözővé tehetünk a szimbólumok jelölésével). A generált automata determinisztikus. A generált automatából eltávolíthatjuk a jelöléseket (ha voltak) és egyesíthetjük azokat az állapotokat amelyeket ugyanahhoz a (jelöletlen) szimbólumhoz társított a konstrukció, így egy nemdeterminisztikus automatát kapunk, amely éppen a jelöletlen reguláris kifejezés által előállított nyelvet fogadja el.

3.4 Definíció (Folyomány)
Legyen $E \in RE$ páronként különböző szimbólumokból álló reguláris kifejezés, akkor az $\alpha \in E$ szimbólum folyományának nevezzük azon deriváltakat, amelyekre $(\omega\alpha)^{-1} E \neq 0$ ahol $\omega \in \Omega^*$.

E egy adott szimbólumához tartozó folyományok ekvivalensek a 3.1. Tétel értelmében (lásd [8]), így minden szimbólumhoz egy egyértelmű folyományt (ekvivalencia osztályt) rendelhetünk. Az alábbi algoritmus a $L(E)$ nyelvet elfogadó automatát hoz létre a szimbólumok folyományaiból, úgy, hogy E minden szimbólumának megfelel az automata egy állapota (amely az adott szimbólumhoz tartozó folyomány osztályt reprezentálja).

3.2. Algoritmus. *Berry-Sethi konstrukció [8].*
Input: páronként különböző szimbólumokból álló reguláris kifejezés az Ω ábécé felett,
Output: az L(E) nyelvet elfogadó M(E) determinisztikus véges automata.

1. *az E reguláris kifejezés minden szimbóluma folyományához társítva tartalmaz M(E) egy állapotot, azonkívül egy megkülönböztetett (kezdő) állapotot magához a teljes reguláris kifejezéshez rendelve.*

2. *A p állapotból a q állapotba vezessen egy átmenet az α szimbólummal címkézve (a q állapot az α szimbólum folyományához tartozik) akkor és csak akkor, ha a p állapot valamely C folyományhoz van rendelve és a C folyomány elő tud állítani az α szimbólummal kezdődő sorozatot.*

3. *Egy állapot akkor és csak akkor elfogadó állapot, ha egy olyan C folyományhoz van rendelve, amelyre $\Delta(C) = 1$ teljesül.*

Berry-Sethi algoritmusának bonyolultsága kvadratikus: az eljárás létrehoz egy determinisztikus elfogadó automatát amelyben az átmenetek száma legfeljebb kvadratikus [44], mindezt kvadratikus számítási időben (beleértve a szimbólumok esetleges jelölését és a jelölések eltávolítását) [13] az input reguláris kifejezés méretéhez (szimbólumai számához) viszonyítva.

3.5 Definíció (Reguláris nyelv véges automatája)
Legyen L reguláris nyelv, amelyet

1. *vagy egy G grammatika*

2. *vagy egy $E \in RE$ reguláris kifejezés*

generál, akkor az L-hez rendelt véges automatának nevezzük és $M(L)$-el jelöljük azt az L-et elfogadó véges automatát, amelyet az 1. esetben a 3.1. Algoritmus konstruál ($M(L) = M(G)$), a 2. esetben a 3.2. Algoritmus hoz létre ($M(L) = M(E)$).

3.1. Példa. *Legyen $G = (\{S, A, B\}, \{a, b\}, S, P)$ reguláris grammatika, ahol $P = \{S \Rightarrow aS, S \Rightarrow bS, S \Rightarrow aA, A \Rightarrow bB, B \Rightarrow a\}$.*

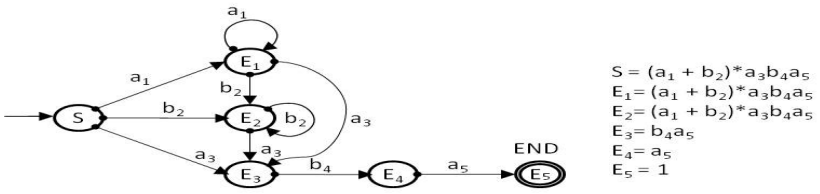
Az $E = (a+b)^* aba$ reguláris kifejezés szintén az $L(G)$ reguláris nyelvet állítja elő. A 6. ábra mutatja a determinisztikus véges automatát amelyet a 3.2. Algoritmus konstruál, E-ben a szimbólumokat jelöléssel megkülönböztetve. A megfelelő nemdeterminisztikus automata a 7. ábrán látható.

3.2 Megjegyzés
A 3.2. Algoritmussal előállított automata gráfjának csúcsai (az automata állapotai) megfelelnek az input reguláris kifejezés szimbólumainak: a konstrukció biztosítja, hogy minden csúcsnál a bemenő élek azonos szimbólummal vannak címkézve, és mindegyik szimbólum pontosan egy csúcshoz legalább egy bemenő élt címkéz. (Feltételezzük, hogy az input reguláris kifejezésben előforduló szimbólumok páronként különbözőek, ha nem, jelöléssel különbözővé kell

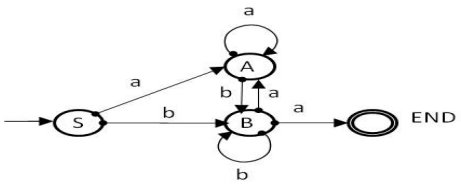
tenni őket). Azt látjuk, hogy ez a konstrukció megfelel a duális nyelv előzőekben megadott definíciójának: a reguláris kifejezés előállít egy reguláris nyelvet, és a konstruált automata állapotai ezen nyelv duális nyelve szimbólumai. Azért, hogy a céljainknak megfelelő automatát kapjunk, a 3.2. konstrukciót a reguláris nyelv szimbólumhalmazait reprezentáló (a szabálysémákban használt szimbólumokhoz hasonló) szimbólumokból alkotott reguláris kifejezésre kell alkalmazni. Az elfogadott mondatok lesznek a kiterjesztett reláció „sémamondatai": az eredeti (alap)nyelv mondatait, a reguláris adatbázis sorait értékbehelyettesítéssel kapjuk meg a „sémamondatokból", úgy, hogy a dom értéktartomány elemeit rendeljük a reguláris kifejezés szimbólumaihoz.

Az XML sémanyelvek (jelölő nyelvek) elemnevekből képzett reguláris kifejezésekkel deklarálnak elemtípusokat (duális nyelv), a megfelelő XML dokumentumok tartalmazzák az elemnevekhez rendelt értékeket, a 7.3. példa szerint. A jelölők nyelve nyilván az értékek duális nyelve, de a jelölőket felfoghatjuk az értéktípusok csoportszimbólumaiként is.

Van egy látszólagos ellentmondás a 3.1. Algoritmus és a 3.2. Algoritmus által konstruált automaták „szemantikája" között: a 3.1. Algoritmus automatája az A csúcsból kifutó éleken helyezi el az A típushoz tartozó értékeket, míg a 3.1. Algoritmus automatájánál megfordítva: az A csúcsba tartó élek kapják az A típushoz tartozó értékeket. A két forma nyilván ekvivalens: a 3.1. Algoritmus esetében a nyelvet egy $P_1 : S \Rightarrow sA, A \Rightarrow aB, \ldots, X \Rightarrow x$*-hez hasonló szabályokat tartalmazó grammatika generálja (ahol* s *kezdő, határoló jel,* $s = = \langle$ *-t használtuk, és a szabályokhoz hozzáfűztük az* $X \Rightarrow x\mathcal{ENDE}, \mathcal{ENDE} \Rightarrow\rangle$ *befejezést). A 3.1. Algoritmus esetében a nyelvet generálhatja egy* $P_2 : S \Rightarrow aA, A \Rightarrow bB, \ldots, W \Rightarrow x\mathcal{ENDE}, \mathcal{ENDE} \Rightarrow\rangle$*-hez hasonló szabályokat tartalmazó grammatika (hozzávehetjük a* $\mathcal{START}$ *új kezdőelemet és a* $\mathcal{START} \Rightarrow \langle S$ *kezdő szabályokat).*



6. ábra. Determinisztikus automata gráfja (3.1. Példa), jelölt szimbólumokkal



7. ábra. Nemdeterminisztikus automata gráfja (3.1. Példa), jelöletlen szimbólumokkal

4. Reguláris nyelv duális nyelve és attribútumai

Ebben a fejezetben megadjuk a duális nyelv és a kiterjesztett reláció definícióját. Bevezetjük és formálisan definiáljuk a reguláris nyelv mondatainak típusait és azok szerkezetét (attribútumok).

Kezdjük a 2. Fejezetben informálisan bevezetett duális nyelv és a kiterjesztett reláció formális definiálásával.

4.1. Duális nyelv és kiterjesztett reláció

4.1 Definíció (Duális nyelv)
Legyen L reguláris nyelv amelyet az $M(L)$ véges automata fogad el. Az L nyelv duális nyelvének ábécéjét az $M(L)$ automata állapotai alkotják (jelölése $\Phi(L)$), a $D(L)$ duális nyelv mondatai az állapotszimbólumok sorozatai: az $M(L)$ automata gráfjának az L nyelv mondatait elfogadó bejárásai (START-tól END-ig). Ha $t \in L$ és $\mathrm{w} \in D(L)$ a t-t elfogadó bejárás, akkor azt mondjuk, hogy t típusa w, *azaz* $\mathrm{w} = type(t)$.

A $D(L)$ duális nyelv mindegyik mondata egy sortípust határoz meg; ezen típusok összessége a „kiterjesztett reláció" sémája. A kiterjesztett reláció egy I előfordulása egy rögzített típushoz (duális mondathoz) tartozó kiterjesztett sorok egy halmaza.

4.2 Definíció (Kiterjesztett reláció)
Legyen L reguláris nyelv amelyet az $M(L)$ automata fogad el, legyen $D(L)$ a társított duális nyelv.
$R(L) = \{t | t \in L, type(t) \in D(L)\}$ kiterjesztett reláció L felett. Az R kiterjesztett reláció sémája a társított duális nyelv, azaz $schema(R) = D(L)$. Legyen $\mathrm{w} \in D(L)$ egy duális mondat, akkor az $I_{\mathrm{w}}(L) = \{t | t \in L, type(t) = \mathrm{w}\}$ (vagy egyszerűen I_{w} ha L, vagy I ha w *is egyértelmű a szövegkörnyezetben) sorhalmaz $R(L)$-nek egy előfordulása.*
A továbbiakban jelöljük $\mathrm{w} = schema(I)$*-vel az I előfordulás sémáját (sorainak azonos* w *típusát), és $\mathcal{I}(L) = \{I_{\mathrm{w}_1}, I_{\mathrm{w}_2}, \ldots; \mathrm{w}_i \in D(L)\}$-vel az előfordulások halmazát L felett. $\mathcal{I}(L)$ akkor és csak akkor véges ha az L nyelv nem-rekurzív. Ha $t \in I \in \mathcal{I}(L)$, akkor $type(t) = schema(I) \in schema(R)$.*

4.1 Megjegyzés
Az előző Definícióban $I_{\mathrm{w}}(L)$ a w *duális mondathoz tartozó maximális előfordulás, valamennyi a* w*-nek megfelelő reguláris mondatot tartalmazza. A relációs adatbázisok előfordulás fogalmának jobban megfelel ezen maximális halmaz egy véges részhalmazának a választása. Miután a 2.1. Megjegyzés szerint $I_{\mathrm{w}}(L)$ véges a* w *duális mondathoz tartozó valamely J előforduláson*

a továbbiakban $I_{\mathrm{w}}(L)$ egy tetszőleges részhalmazát fogjuk érteni, azaz $J \subseteq$ $\subseteq \{t | t \in L, type(t) = \mathrm{w}\}$

4.2 Megjegyzés
A 2.1. Megjegyzést megismételve itt is rögzítjük, hogy az L reguláris nyelv mint kiterjesztett reláció előfordulásait egy-egy duális mondathoz tartozó reguláris mondatokra (azok egy részhalmazára) korlátoztuk, erre azért van szükség, hogy a kiterjesztett reláción értelmezett függőségek ellenőrzésénél azonos szerkezetű sorokkal dolgozhassunk és így elkerülhessük, többek között, a hiányos (null) adatokon való függőségvizsgálatot. Az előfordulás fogalmának ezt a leszűkítését a 10.2. Fejezetben M2 megkötésnek nevezzük (és elemezzük feloldásának következményeit). Ez a megkötés véges bemenő ábécé (= véges sok átmenet; az állapotok száma mindig véges) esetén az előfordulás sorainak számát is végessé korlátozza, ezért a szokásos kikötés (az előfordulás a sorok valamely véges halmaza) elhagyható (annak ellenére, hogy a rekurzív L nyelv végtelen). Végtelen bemenő ábécék esetével a 10.5. Fejezetben foglalkozunk.

A fenti Megjegyzés megalapozza a következő egyszerű állítást:

4.1 Lemma
Ha $R(L)$ kiterjesztett reláció az L reguláris nyelv felett és $I \in \mathcal{I}(L)$, akkor I az L nyelv véges részhalmaza.

4.3 Megjegyzés
Itt és a következőkben egy reguláris nyelvet rekurzívnak nevezünk, ha az elfogadó automata gráfja tartalmaz kört (nem-rekurzívnak, ha a gráf körmentes). Ez a szóhasználat alapvetően eltér a szokásostól: rekurzívnak szokás nevezni egy nyelvet, ha azt egy Turing-gép elfogadja, azaz mind a nyelv, mind a komplementere rekurzívan fölsorolható. Mi azért választottuk ezt a szokatlan szóhasználatot, mert a véges automata gráfjában (a nyelvet leíró reguláris kifejezésben) meglévő ciklikusságra szeretnénk utalni.

4.2. A kiterjesztett reláció attribútumai

A relációs modellben a függőségek (tartalmazási, többértékű, funkcionális, összekapcsolási) szintaktikus leírásához relációs attribútumok halmazait használjuk. A reguláris nyelvre alapozott modellben is így fogunk eljárni, ehhez először meg kell határoznunk az attribútumok fogalmát, valamint azt is, miként tudunk egy séma attribútumaiból a függőségek számára alkalmas részhalmazokat kiválasztani.

4.3 Definíció (Reguláris attribútum)
Legyen L reguláris nyelv amelyet az $M(L)$ véges automata fogad el. Legyen $D(L)$ az L duális nyelve és legyen $\mathrm{w} \in D(L)$ egy nem üres duális mondat ($\mathrm{w} \neq \epsilon$). Legyen továbbá $\mathrm{w} = v_1\, v_2\, \ldots\, v_n; v_i \in \Phi(L), 1 \leq i \leq n$. A $v_i, 1 \leq i \leq n$ (nem feltétlenül különböző, de sorrendjük szerint megkülönböztetett) szimbólumokat a w *duális mondathoz, mint sortípushoz tartozó reguláris mondatok (sorok) attribútumainak nevezzük. Ha $t \in I_{\mathrm{w}}$ ($type(t) = schema(I_{\mathrm{w}}) = \mathrm{w}$) és $t = t_1\, t_2\, \ldots\, t_n$, akkor $t[v_i] = t_i$ a t sor v_i attribútumon felvett értéke, vagy, a v_i attribútum a t sorból a t_i értéket vetíti ki.*

A reguláris nyelven értelmezendő függőségek szintaktikus leírásához szükséges attribútumhalmazok kiválasztásához meg kell különböztetnünk két esetet:

1. nem-rekurzív nyelv
2. rekurzív nyelv

Ha az L nyelv nem rekurzív, azaz ha az $M(L)$ automata gráfja körmentes (a generáló reguláris grammatika / kifejezés nem-rekurzív), akkor a társított $D(L)$ duális nyelv véges, mivel az $M(L)$ gráfnak véges sok bejárása van (azonos jelölést - $M(L)$ - használunk a véges automatára és gráf reprezentációjára). Ezen duális mondatok mindegyikén kijelölhetünk a függőségeket specifikáló attribútumhalmazokat.

Ha az $M(L)$ automata gráfja tartalmaz kört (a generáló reguláris grammatika / kifejezés rekurzív), akkor a társított $D(L)$ duális nyelv végtelen. Használhatjuk a pumpálási eljárást arra, hogy kiválasszunk részsorozatokat a duális mondatokból és függőségeket értelmezzünk ezeken a részsorozatokon. Kiválaszthatunk utakat a nem-pumpált részen, azután vehetünk részsorozatokat a pumpált részen és együtt pumpálhatjuk a kiválasztott részeket. Természetesen, a bejárt csúcsokat mindig egy teljes bejárásból kell kiválasztanunk ($\mathcal{START}$-tól $\mathcal{END}$-ig) a bejárás sorrendjében, és az ismétlődések is a bejárás sorrendjében szerepelnek, mivel a (bejárás által generált) duális mondat a definiálandó függőség értelmezési tartománya.

Megjegyezzük, hogy a függőség szintaktikus definícióját a $D(L)$ duális nyelven adjuk meg; a függőség szemantikáját (a kielégítettség ellenőrzését) az L nyelv mondatain értelmezzük.

El szeretnénk kerülni, hogy a definiálandó függőséget minden egyes (esetleg végtelen sok) duális mondaton meg kelljen adnunk: ehelyett az $M(L)$

gráfon specifikáljuk a függőséget, és ezt a specifikációt alkalmazzuk a duális mondatokra (kiválasztás). Abból a célból, hogy a függőséget specifikálhassuk, kijelölhetünk csúcsokat és kimondhatjuk, hogy egy bejárás során minden áthaladás kiválasztja őket. Kijelölhetünk kezdő és befejező csúcsokat a bejárásból, úgy, hogy ez a két csúcs az útvonal minden záródásánál kerüljön kiválasztásra.

4.4 Definíció (Kijelölés)
Legyen L reguláris nyelv és legyen $M(L)$ a nyelvet elfogadó véges automata gráfja. Az $M(L)$ feletti Y kijelölésnek nevezzük az (Y_1, Y_2) párost, ahol $Y_1 \subseteq \subseteq nodes(M(L))$ és Y_2 az $M(L)$ gráf tranzitív lezártjának egy részgráfja. Y_1-et az $M(L)$ gráf azon csúcsaiból vesszük, amelyek nem tartoznak körhöz, Y_2 pedig olyan csúcsokat tartalmaz amelyek egy bejárás során ismétlődően szerepel(het)nek.

4.4 Megjegyzés
Legyen $X=(X_1, X_2)$ kijelölés egy relációs típusú grammatikával generált reguláris nyelv elfogadó véges automatája felett, akkor X_2 üres gráf, amint a 6.2. példában is látható. Ez a tény a 3.1. Megjegyzésből is következik.

Egy kijelölés egy adott duális mondatból egyértelműen kell, hogy kiválasszon egy részsorozatot. Megadunk két különböző, az alábbiakban részletezett módszert amelyekkel a kijelölés (a módszerre nézve egyértelműen) kiválaszt egy részsorozatot.

Legyen $\mathrm{w}=v_1v_2\ldots v_n$ duális mondat az $M(L)=(V,E)$ gráf felett, és jelölje $walk(\mathrm{w}) = (\mathcal{START}, v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n, \mathcal{END})$ (rövidítve $walk(\mathrm{w}) = = (v_1, v_2, \ldots, v_n)$) $M(L)$-nek a w-t előállító bejárását.

4.5 Definíció (Szigorú kiválasztás)
Legyen $Y = (Y_1, Y_2)$ egy kijelölés és w *duális mondat $M(L)$ felett. Legyen $walk(\mathrm{w}) = (v_1, v_2, \ldots, v_n)$. Y_1 szimbólumai bejárásuk szerint kerülnek kiválasztásra (ha a bejárás érinti őket). Minden $e \in E(Y_2)$ él esetén, az él végpontjai az elérés sorrendjében kerülnek kiválasztásra (ha a bejárás érinti őket) a $walk(\mathrm{w})$ szerinti bejárás során, mindannyiszor, amikor mindkét végpont sorra kerül. Azaz, ha az e él két végpontja A és B és ha $A = v_i, B = v_j$ valamely $1 \le i < j \le n$-re, akkor v_i és v_j akkor és csak akkor választódik ki, ha $v_k \neq A, B; i < k < j$. Y_2 izolált csúcsait minden áthaladásnál (ha van ilyen) kiválasztjuk a $walk(\mathrm{w})$ által megadott bejárás során. A kiválasztást Y_2 minden élére és izolált csúcsára egymástól függetlenül elvégezzük. A kiválasztási folyamat végére a* w *duális mondatból kiválasztott szimbólumokból felépül az (esetleg üres)* $\mathrm{w}[Y] = v_{i_1}v_{i_2}\ldots v_{i_k}$ *$(1 \le i_1 < i_2 < \ldots < i_k \le n\,(k \ge 0))$ sorozat.*

Legyen $t \in L$, a megfelelő duális mondat $\mathrm{w} \in D(L)$. A $\mathrm{w}[Y]$ szimbólumsorozatot interpretálhatjuk úgy, mint egy (nem szükségképpen különböző) „attribútumokból" álló tömböt (array), amely a t kiterjesztett „sor"-t az értékek (tömb index szerint) megfelelő $t[Y]$ tömbjére vetíti. Ha $\mathrm{w}[Y] = \epsilon$, akkor $t[Y] = \epsilon$ úgyszintén (ϵ jelöli az üres szimbólumsorozatot).

4.6 Definíció (Teljes kiválasztás)
Legyen $Y = (Y_1, Y_2)$ egy kijelölés és w *duális mondat $M(L)$ felett. Legyen $walk(\mathrm{w}) = (v_1, v_2, \ldots, v_n)$. Y_1 szimbólumai bejárásuk szerint kerülnek kiválasztásra (ha a bejárás érinti őket). Minden $(A, B) \in E(Y_2)$ él esetén, az A végpont minden olyan érintés esetén kiválasztásra kerül, ha ezt követően a B csúcs is sorra kerül a $walk(\mathrm{w})$ bejárás során, és a B csúcs minden olyan érintésnél kiválasztásra kerül, ha az A csúcsot követi. Az (A, B) él nem indukálja az A csúcs kiválasztását a bejárás során, ha a B csúcs a bejárás további részében nincs jelen, és hasonlóképpen, a B kiválasztására nem kerül sor, ha a bejárásban őt megelőzően az A csúcs nem szerepelt. A teljes kiválasztás azt jelenti, hogy az elsőként bejárt A csúcs és az utolsóként bejárt B csúcs között valamennyi A és B csúcs kiválasztásra kerül. Y_2 izolált csúcsait minden áthaladásnál (ha van ilyen) kiválasztjuk a $walk(\mathrm{w})$ által megadott bejárás során. A kiválasztást Y_2 minden élére és izolált csúcsára egymástól függetlenül elvégezzük. A kiválasztási folyamat végére a* w *duális mondatból kiválasztott szimbólumokból felépül az (esetleg üres)* $\mathrm{w}\{Y\} = v_{i_1} v_{i_2} \ldots v_{i_k}$ *($1 \leq i_1 < i_2 < \ldots < i_k \leq n\ (k \geq 0)$).*

Legyen $t \in L$, a megfelelő duális mondat $\mathrm{w} \in D(L)$. A $\mathrm{w}\{Y\}$ szimbólumsorozatot interpretálhatjuk úgy, mint egy (nem szükségképpen különböző) „attribútumokból" álló tömböt (array), amely a t kiterjesztett „sor"-t az értékek (tömb index szerint) megfelelő $t\{Y\}$ tömbjére vetíti. Ha $\mathrm{w}\{Y\} = \epsilon$, akkor $t\{Y\} = \epsilon$ úgyszintén (ϵ jelöli az üres szimbólumsorozatot).

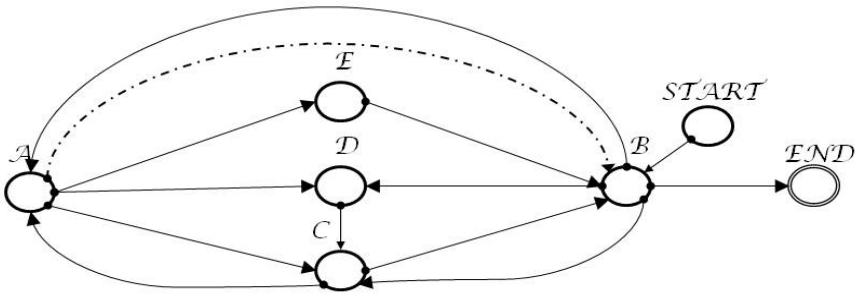
4.1. Példa. *Legyen $M(L)$ a 8. ábrán adott gráf. Legyen Y egy kiválasztás $M(L)$ felett amely csak az (A, B) élből áll. Legyen*

$$BCADCACAEBDCBAEB.$$

$M(L)$ egy bejárása. A szigorú kiválasztás ($\mathrm{w}[Y]$) kiválasztja a kisbetűvel jelzett csúcsokat a bejárás sorrendje szerint alsó indexben számozva:

$$BCADCACa_1Eb_1DCBa_2Eb_2.$$

A teljes kiválasztás ($\mathrm{w}\{Y\}$) kiválasztja az (A, B) él csúcsait mindannyiszor amikor a bejárás során sorra kerülnek; a kiválasztott kisbetűvel jelzett csúcsokat a (rekurzív) bejárás sorrendjében számoztuk:



8. ábra. Véges automata gráfja és a tranzitív lezáró élek

$$BCa_1DCa_2Ca_3Eb_1DCb_2a_4Eb_3.$$

Az összefoglaló táblázat mutatja a két kiválasztás eltérését:

```
          1234567890123456
w=        BCADCACAEBDCBAEB
w[Y]=             A B   A B
w{X}=       A  A A B  BA B
```

azaz, $\mathrm{w}\,[Y] = ABAB$ *és* $\mathrm{w}\,\{Y\} = AAABBBAB$.
A teljes kiválasztás a 3,6 pozíciókon is kiválasztja az A*, a 13. pozíción a* B *csúcsot, a szigorú kiválasztás viszont nem, mert az* (A, B) *(tranzitív lezárthoz tartozó) él által adott feltétel nem teljesül.*

5. Összehasonlító eljárások kiterjesztett relációkon

A függőségek szemantikájához értelmeznünk kell az attribútumok (attribútumhalmazok) kiterjesztett sorokon felvett értékeit. A relációs modellben egy attribútum egy adott soron felvett értékét a relációs algebra projekció művelete „vetíti ki" a sorból. Egy relációs attribútumhalmaz két soron felvett értékeinek összehasonlításakor az egyes attribútumoknak a két soron felvett értékeit hasonlítjuk páronként össze. A kiterjesztett reláció attribútumai a duális nyelv szimbólumai, az attribútum egy kiterjesztett soron felvett értéke a sor azon pozícióján álló szimbólum (az alap reguláris nyelv ábécéjéből) amely pozíción az attribútumot reprezentáló duális szimbólum áll a duális mondatban (4.3. Definíció). Azt mondhatjuk, hogy a sorból ezt az értéket az adott attribútum „vetíti ki". A kiterjesztett reláció attribútumhalmazait az előző fejezetben definiált *kijelölés* határozza meg: a *kijelölés* egy duális mondatból kiválaszt attribútumokat (nem szükségképpen különböző duális szimbólumokat, ezek a szimbólumok a duális mondatbeli pozíciójukkal együtt reprezentálják az attribútumokat), ezek az attribútumok egy előfordulás soraiból (a reguláris nyelv mondataiból) jelsorozatokat vetítenek ki, a sorból kivetített jelsorozat az attribútumhalmaz adott soron felvett értéke.
A továbbiakban a szigorú kiválasztás (4.5. Definíció) szintaktikáját használjuk, de az elmondottak a teljes kiválasztásra (4.6. Definíció) is érvényesek.

5.1 Definíció (Projekció)
Legyen L reguláris nyelv, Y egy kijelölés $M(L)$ felett. Legyen $t \in L$, a megfelelő duális mondat $\mathrm{w} \in D(L)$ *($\mathrm{w} = type(t)$). A* $\mathrm{w}[Y]$ *szimbólumsorozat az Y kijelölés interpretációja a* w *típuson (duális mondaton). Ha I egy előfordulás, amelyre $schema(I) = \mathrm{w}$, akkor azt is mondhatjuk,* $\mathrm{w}[Y]$ *a $schema(I)$ (rendezett) attribútumhalmaza. A* $\mathrm{w}[Y]$ *attribútumhalmaz a t kiterjesztett „sor"-t az értékek (tömb index szerint) megfelelő $t[Y]$ tömbjére vetíti. Ha* $\mathrm{w}[Y] = \epsilon$, *akkor $t[Y] = \epsilon$ úgyszintén (ϵ jelöli az üres szimbólumsorozatot).*

5.1 Megjegyzés
A továbbiakban mindig feltételezzük, hacsak nem jelezzük másként, hogy a $\mathrm{w}[Y]$ *attribútumhalmaz szimbólumok pozíción megkülönböztetett, pozíción rendezett sorozata, így például $AAA \neq AAAA; AAAB \neq AABA$.*

5.2 Definíció (Projekciók szigorú összehasonlítása)
Legyen L reguláris nyelv, Y egy kijelölés $M(L)$ felett. Legyenek $t^1, t^2 \in L$ úgy, hogy $type(t^1) = type(t^2) = \mathrm{w}$. Legyen $\mathrm{w}[Y] = v_1 v_2 \ldots v_k; k \geq 0$ *és $t^i[Y] = t^i_1\, t^i_2\, \ldots\, t^i_k; i = 1,2$. Azt mondjuk, hogy $t^1[Y]$ (szigorúan) egyenlő $t^2[Y]$-nal*

(jelölve $t^1[Y] = t^2[Y]$*), ha* $t^1_1 = t^2_1, t^1_2 = t^2_2, \ldots, t^1_k = t^2_k$*. Ha* $\mathrm{w}[Y] = \epsilon$*, akkor* $t^1[Y] = t^2[Y]$.

A projekciók szigorú összehasonlítása a formális nyelvek világában szokásos jelsorozat-egyenlőség: két jelsorozat egyenlő, ha azonos hosszúságúak és a megfelelő pozíciókon azonos szimbólumokat tartalmaznak. A projekció szimbólumainak típusa a kivetítő attribútum: egy ismétlődő attribútum kivetíthet azonos szimbólumokat egy sorból (az ismétlődő attribútum értékei között a soron lehetnek azonosak). Értelmezhetünk olyan összehasonlításokat, amelyek az ismétlődő értékeket különböző módon kezelik, ilyen módon az összetett értékű adatbázisokon értelmezett függőségek világába jutunk (10.3. Fejezet).

6. Funkcionális függőség a duális nyelven definiálva

Legyen L reguláris nyelv, legyen $M(L)$ az elfogadó automatája, legyen $D(L)$ a társított duális nyelv. Kiválaszthatunk két részsorozatot mindegyik duális mondaton, mint a függőség bal és jobb oldalát, tekintsük ezt a függőség szintaktikus specifikációjának. Az I előfordulás (az L reguláris nyelv mondatainak halmaza) kielégíti ezt a függőséget, ha nincs két olyan sor I-ben amelyek azonosak a bal oldali részsorozat duális nyelvi szimbólumaihoz illeszkedő szimbólumokban, de legalább egy jobb oldali duális nyelvi szimbólumhoz illeszkedő szimbólumban különböznek.

A funkcionális függőség szintaktikus definícióját a $D(L)$ duális nyelven adjuk meg; a függőség szemantikáját (a kielégítettség ellenőrzését) az L nyelv mondatain értelmezzük.

Ha az $M(L)$ automata gráfja körmentes, akkor a társított $D(L)$ duális nyelv véges, mivel az $M(L)$ gráfnak véges sok bejárása van. Ezen duális mondatok mindegyikén definiálhatunk funkcionális függőséget (bal és jobb oldalt) és vizsgálhatjuk ezek logikai implikációját.

Ha az $M(L)$ automata gráfja tartalmaz kört, akkor a társított $D(L)$ duális nyelv végtelen. Használhatjuk a pumpálási eljárást arra, hogy kiválasszunk részsorozatokat a duális mondatokból és funkcionális függőségeket értelmezzünk ezeken a részsorozatokon. Kiválaszthatunk utakat a nem-pumpált részen, azután vehetünk részsorozatokat a pumpált részen (a bal és a jobb oldal számára) és együtt pumpálhatjuk a kiválasztott részeket. Természetesen, a bejárt csúcsokat mindig egy teljes bejárásból kell kiválasztanunk ($\mathcal{START}$-tól $\mathcal{END}$-ig) a bejárás sorrendjében, és az ismétlődések is a bejárás sorrendjében szerepelnek, mivel a (bejárás által generált) duális mondat a funkcionális függőség értelmezési tartománya.

A kiválasztási eljárás ismételt alkalmazásával két részsorozatot kell kiválasztanunk minden duális mondatból: egyet a funkcionális függőség bal oldala, egyet a jobb oldala számára.

El szeretnénk kerülni, hogy a funkcionális függőséget minden egyes (esetleg végtelen sok) duális mondaton meg kelljen adnunk: ehelyett az $M(L)$ gráfon specifikáljuk a funkcionális függőséget, és ezt a specifikációt alkalmazzuk a duális mondatokra (kiválasztás). Abból a célból, hogy a funkcionális függőség oldalait (jobb és bal) specifikálhassuk, a $M(L)$ gráf csúcsainak két

halmazát kell kijelölnünk: egy halmazt a bal (jelöljük ezt X-szel), egy másikat a jobb oldal számára (jelöljük ezt Y-nal). Kijelölhetünk csúcsokat és kimondhatjuk, hogy egy bejárás során minden áthaladás kiválasztja őket. Kijelölhetünk kezdő és befejező csúcsokat a bejárásból, úgy, hogy ez a két csúcs az útvonal minden záródásánál kerüljön kiválasztásra.

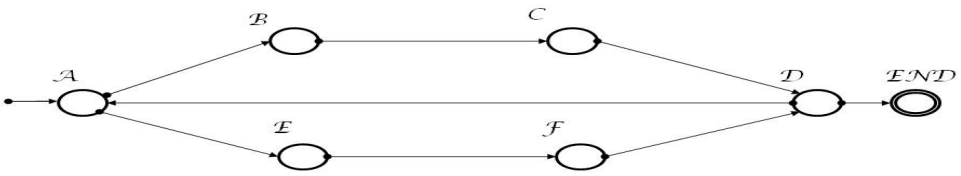
A szigorú és teljes kiválasztás különböző módozatokat nyújt arra, hogy a funkcionális függőségekben a reguláris kifejezések ismétlődő részeit figyelembe vehessük. Mindkét kiválasztási módszert két további módon alkalmazhatjuk a funkcionális függőség szintaktikus leírásánál (a két változat ismertetésekor a szigorú kiválasztás jelölésmódját használjuk, de a változatok a teljes kiválasztásra is alkalmazhatóak):

6.1 Definíció (A változat: kétfázisú kiválasztás)
Legyen w *duális mondat és legyen* $X = (X_1, X_2)$ *és* $Y = (Y_1, Y_2)$ *két kijelölés* $M(L)$ *felett úgy, hogy* $X_1 \subseteq Y_1$ *és* $X_2 \subseteq \overline{Y}_2$ *(ahol* $\overline{Y}_2$ *az* Y_2 *tranzitív lezártját jelöli). A kiválasztási eljárás amely* $\mathrm{w}[Y]$*-t létrehozza egyidejűleg bejárja* X *éleit és csúcsait is: az így kiválasztott szimbólumsorozatot* $\mathrm{w}[Y[X]]$*-szel jelöljük.*

Legyen $t \in L$, a megfelelő duális mondat $\mathrm{w} \in D(L)$. A $\mathrm{w}[Y[X]]$ szimbólumsorozatot interpretálhatjuk úgy, mint egy (nem szükségképpen különböző) „attribútumokból" álló tömböt (array), amely a t kiterjesztett „sor"-t az értékek (tömb index szerint) megfelelő $t[Y[X]]$ tömbjére vetíti.

6.2 Definíció (B változat: Egyfázisú vagy egyszerű kiválasztás)
Legyen w *duális mondat és legyen* $X = (X_1, X_2)$ *és* $Y = (Y_1, Y_2)$ *két kijelölés* $M(L)$ *felett. Egyfázisú kiválasztást végzünk* X *és* Y *alapján, ha a* $\mathrm{w}[X]$ *és* $\mathrm{w}[Y]$ *kiválasztások egymástól függetlenül történnek.*

6.1. Példa. *Nézzük meg a különbséget az egyszerű és a kétfázisú (w[Y[X]]) kiválasztás között. A 9. ábrán látható az* A, B, C, D, E, F *csúcsok és* (A, B), (B, C), (C, D), (D, A), (A, E),(E, F), *valamint* (F, D) *élek gráfja.*
Legyen
$Y = (\{\}, (\{A, B, C, D\}, \{(A, B), (B, C), (C, D)\}))$ *és*
$X = (\{\}, (\{A, D\}, \{(A, D)\}))$
két kijelölés és legyen $\mathrm{w} = ABCDABCDAEFDABCD$ *egy duális mondat. Jelöljük a különböző* $\mathrm{w}[Y]$, $\mathrm{w}[Y[X]]$ *és* $\mathrm{w}[X]$ *módszerekkel kiválasztott csúcsokat kisbetűvel, majd csak a kiválasztott csúcsokat, végül a kiválasztott jelsorozatokat összefűzve:*



9. ábra. Egyszerű véges automata gráfja

```
          1234567890123456
w[Y]=     abcdabcdAEFDabcd
          ABCDABCD    ABCD
w[Y[X]]= aBCdaBCdAEFDaBCd
          A   DA  D    A
w[X]=     aBCdaBCdaEFdaBCd
          A   DA  DA  DA  D
```

azaz,
$\mathrm{w}[Y] = ABCDABCDABCD,$
$\mathrm{w}[Y[X]] = ADADA$ *és*
$\mathrm{w}[X] = ADADADAD.$

A bejárás egy adott pozícióján kiválasztott csúcsok különböznek az egyes kiválasztási módszerek szerint: $\mathrm{w}[X]$ *a 9-es bejárási pozícióra az A csúcsot választja ki (jelölése a), míg* $\mathrm{w}[Y[X]]$ *a 9-es pozíción nem választ ki semmit (jelölése A). A különböző kiválasztást az okozza, hogy az* Y_2 *gráf nem tartalmazza az* (A, E), (E, F) *és* (F, D) *éleket, így az* Y_2*-ben nem, csak a tranzitív lezártjában szereplő* (A, D) *él végpontjai csak akkor választhatóak ki ha őket az* (A, B), (B, C), (C, D) *éleket bejárva érjük el. A kisbetűs jelöléssel megfogalmazva ez azt jelenti, hogy az* X*-beli él végpontjait csak akkor vehetjük figyelembe, ha olyan csúcsokon járunk, amelyeket* $\mathrm{w}[Y]$*-ban kisbetűvel jelöltünk.*

6.1. Reguláris funkcionális függőség

A reguláris funkcionális függőség definíciójában a 6.2. Definícióban (B változat: Egyfázisú vagy egyszerű kiválasztás) megadott kiválasztási módszert

használjuk.

6.3 Definíció (Reguláris funkcionális függőség)
Legyen L reguláris nyelv és legyen $M(L)$ az L nyelvet elfogadó véges automata gráf reprezentációja. Legyen $X=(X_1,X_2)$ és $Y=(Y_1,Y_2)$ két kijelölés $M(L)$ felett. Az $M(L)$ felett értelmezett funkcionális függőségnek (reguláris FD; RFD) nevezzük az $X \to Y$ kifejezést. Az $I \in \mathcal{I}(L)$ (véges) adatbázis előfordulás kielégíti az $X \to Y$ funkcionális függőséget (jelölve $I \models X \to Y$), ha bármely két $t_1, t_2 \in I$ sorra $t_1[X] = t_2[X]$ csak akkor teljesülhet, ha $t_1[Y] = t_2[Y]$ is igaz. Az $Y = M(L)$ esetet kulcs függőségnek nevezzük.

6.1 Megjegyzés
Ha a 6.1. Definíció szerinti kiválasztási módszert (A változat: Kétfázisú kiválasztás) szeretnénk használni a funkcionális függőség 6.3. Definíciójában, akkor a $t_1[X] = t_2[X]$ kifejezés helyett a $t_1[Y[X]] = t_2[Y[X]]$ kifejezést kell vennünk. Ebben az esetben meg kell követelnünk, hogy $X_1 \subseteq Y_1$ és $X_2 \subseteq \overline{Y}_2$ teljesüljön.

6.2 Megjegyzés
Azt mondjuk, hogy az L reguláris nyelv gyengén kielégíti a $\sigma = X \to Y$ RFD-t (jelölve $L \overset{w}{\models} \sigma$), ha bármely két $t_1, t_2 \in L$ sor esetén, amelyekre $\mathrm{w} = type(t_1) = type(t_2)$ teljesül, ha a két sor megegyezik X-en, azaz, $t_1[X] = t_2[X]$, akkor $t_1[Y] = t_2[Y]$ is teljesül. Azt mondjuk, hogy az L reguláris nyelv erősen kielégíti a σ RFD-t (jelölve $L \overset{s}{\models} \sigma$), ha bármely két $t_1, t_2 \in L$ sor esetén, amelyekre $\mathrm{w}_i = type(t_i)$, $i=1,2$ és $\mathrm{w}_1[X] = \mathrm{w}_2[X]$ és $\mathrm{w}_1[Y] = \mathrm{w}_2[Y]$ teljesül, mindannyiszor ha a két sor megegyezik X-en, azaz, $t_1[X] = t_2[X]$, akkor $t_1[Y] = t_2[Y]$ is teljesül. Nyilván, ha $L \overset{s}{\models} \sigma$, akkor $L \overset{w}{\models} \sigma$. A következőkben, ha nem jelezzük másként, a funkcionális függőség kielégítését a 6.3. Definícióban specifikált módon értelmezzük.

6.2. Példa. *Legyen R(A,B,C,D,E) egy relációs séma és legyen G=(N,T,S,P) a 2.1. példa szerinti grammatika. Legyen $L(G)$ a generált reguláris nyelv és legyen $X = (\{B,C\}, (\{\}, \{\}))$ és $Y = (\{D\}, (\{\}, \{\}))$ két kijelölés $M(L)$ felett. A relációs esetben a kiválasztási módszerek azonos eredményre vezetnek, mivel a nyelvet elfogadó véges automata gráfja körmentes, ahogy a 10. ábrán is látható (lásd még a 3.1. Megjegyzést).*

Azaz, az $X \to Y$ RFD a 6.3. Definíció szerint specifikálva a hagyományos relációs funkcionális függőséggel azonos: $\{B,C\} \to \{D\}$.

Egy $I \in \mathcal{I}(L)$ előfordulás kielégíti az $X \to Y$ reguláris funkcionális függőséget ha I bármely két olyan mondata esetén amelyeknél a B és C nemterminális szimbólumok helyettesítéseként generálódott terminális szimbólumok

rendre megegyeznek, akkor azon terminális szimbólumai a két mondatnak, amelyek D helyettesítéseként keletkeztek, szintén azonosak.

6.3. Példa. *Nézzünk egy a valós életből vett (egyszerűsített) példát a szigorú és teljes kiválasztás különbözőségének bemutatására. A bemutatott DTD részlet egy (egyszerűsített) elemleírás az NCBI MMDB.dtd DTD sémaleírásából ([41]):*

```
<!ELEMENT Biostruc
   (Biostruc_mmdb_id,
     (Biostruc_local_id,
        (Biostruc_name | Biostruc_comment |
         Biostruc_history | Biostruc_attribution))*)>
```

Ez az elem egy biológiai makromolekula egy komponensének struktúráját írja le. Intuitíve megfogalmazhatunk két függőségi feltételt:

1. *két azonos mmdb_id-jű tételhez a* $(local\_id, attribution)$ *párok azonos listája kell, hogy tartozzék*

2. *két azonos mmdb_id-jű tételhez a local_id-ek azonos listája és a (local_id,attribution) párok azonos listája kell, hogy tartozzék*

Definiáljuk az
$RFD_{Mol} : (\{Biostruc\_mmdb\_id\}, (\{\}, \{\})) \rightarrow$
$(\{\}, (\{Biostruc\_local\_id, Biostruc\_attribution\},$
$\{(Biostruc\_local\_id, Biostruc\_attribution)\}))$
reguláris funkcionális függőséget.
Szigorú kiválasztást használva, RFD_{Mol} *az első megszorítást írja le, míg teljes kiválasztással a második megszorítást fejezi ki.*

6.4. Példa. *Az alábbi példa is a szigorú és teljes kiválasztás hatását demonstrálja a reguláris funkcionális függőség által kifejezett megszorítás tartalmára nézve. Az alábbi DTD részlet laboratóriumi tesztadatok elemleírását adja meg:*

```
<!ELEMENT LabTest
   (TestName,
     (TestStepId, TestStepResult, (NextStep | Summary))*)>
```

Ez az elem laboratóriumi tesztlépések sorozatát írja le. Az utolsót megelőző minden tesztlépés utal egy következőre, az utolsó lépés összegzi a teljes tesztfolyamatot. Intuitíve megfogalmazhatjuk a következő két megszorítást:

1. *két tétel azonos $TestName$-val kell, hogy azonos $(TestStepId, Summary)$ párt tartalmazzon*

2. *két tétel azonos $TestName$-val kell, hogy azonos $TestStepId$-ekből álló listát és azonos $(TestStepId, Summary)$ párt tartalmazzon*

Definiáljuk az
$RFD_{Test} : (\{TestName\}, (\{\}, \{\})) \rightarrow$
$(\{\}, (\{TestStepId, Summary\}, \{(TestStepId, Summary)\}))$
reguláris funkcionális függőséget.
Szigorú kiválasztást használva, RFD_{Test} az első megszorítást írja le, míg teljes kiválasztással a második megszorítást fejezi ki.

6.1. Észrevétel. *Legyen $X = (X_1, X_2)$ és $Y = (Y_1, Y_2)$ két kijelölés az L reguláris nyelv felett, úgy hogy mind X_2, mind Y_2 valamennyi csúcsa izolált, akkor a szigorú és teljes kiválasztás ekvivalens az $X \rightarrow Y$ RFD-nek az L valamely előfordulásán való kielégítettségére vonatkoztatva.*

6.2. Észrevétel. *Legyen $X = (X_1, X_2)$ és Y két kijelölés az L reguláris nyelv felett, úgy hogy X_2 valamennyi csúcsa izolált, akkor minden $I \in \mathcal{I}(L)$ előfordulás esetén, ha $I \models X \rightarrow Y$ a teljes kiválasztással értelmezve, akkor $I \models X \rightarrow Y$ a szigorú kiválasztással is.*

6.2. Reguláris funkcionális függőségek logikai implikációja

6.3 Megjegyzés
A továbbiakban feltételezzük hogy egy L reguláris nyelvhez mindig van egy hozzárendelt, a funkcionális függőségek leírásához és értelmezéséhez használt kiválasztási eljárás a 4.5 - 4.6 Definíciók és a 6.1 - 6.2 közül kijelölve. Ha ez a kiválasztási eljárás a 6.1. Definíció (kétfázisú kiválasztás) módszerét is magában foglalja, akkor az egy RFD-ben foglalt két kijelölésre (legyenek ezek X és Y) az $X_1 \subseteq Y_1$ és $X_2 \subseteq \overline{Y}_2$ tartalmazási relációk kell, hogy teljesüljenek (ahol $\overline{Y}_2$ az Y_2 tranzitív lezártját jelöli).

Vezessünk be néhány hasznos jelölést:

Legyen $X = (X_1, X_2)$ kijelölés $M(L)$ felett, legyen $X_2 = (V, E)$, akkor legyen $nodes(X_2) = V$.

Legyen $X = (X_1, X_2)$ kijelölés $M(L)$ felett, akkor legyen $nodes(X) = = \{X_1\} \cup nodes(X_2)$.

Legyen $\sigma = X \rightarrow Y$ reguláris funkcionális függőség, jelölje $nodes(\sigma)$ az

$M(L)$ azon csúcsainak halmazát, amelyek vagy X-ben vagy Y-ban vannak, azaz, $nodes(\sigma) = nodes(X) \cup nodes(Y)$.
Y-ban vannak, azaz, $nodes(\sigma) = nodes(X) \cup nodes(Y)$.

Legyen $w \in D(L)$ duális mondat, jelölje $nodes(w)$ a w szimbólumainak halmazát. Ha X kijelölés $M(L)$ felett, akkor $nodes(w[X])$ jelöli a $w[X]$ szimbólumainak halmazát.

6.4 Definíció
Legyen L reguláris nyelv és legyen $D(L)$ a társított duális nyelv és legyen $M(L)$ az L nyelv elfogadó véges automatájának gráf reprezentációja. Legyen X kijelölés $M(L)$ felett és legyen $w \in M(L)$ duális mondat. Azt mondjuk, hogy w magában foglalja (subsumes) X-et (jelölve $X \sqsubseteq w$) ha $nodes(X) \subseteq nodes(w)$. Legyen $\sigma = X \to Y$ egy RFD, azt mondjuk, hogy w magában foglalja σ-t (jelölve $\sigma \sqsubseteq w$) ha $X \sqsubseteq w$ és $Y \sqsubseteq w$. Legyen $I \in \mathcal{I}(L)$ egy előfordulás és legyen $w = schema(I)$. I magában foglalja X-et (jelölve $X \sqsubseteq I$) ha $X \sqsubseteq w$. Ha Σ RFD-k egy halmaza, akkor $\Sigma \sqsubseteq I$ akkor és csak akkor, ha $\forall \sigma \in \Sigma$-ra teljesül, hogy $\sigma \sqsubseteq I$.

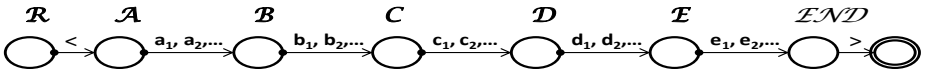
A következő *Feltételt* fogjuk használni ha az L reguláris nyelv függőségeinek egy Γ halmazáról van szó:

$$\text{Létezik egy } I \in \mathcal{I}(L) \text{ előfordulás úgy, hogy } \Gamma \sqsubseteq I \tag{1}$$

6.4 Megjegyzés
A 6.3. Definíció megadja a $M(L)$ gráfon értelmezett funkcionális függőségek szintaktikus formáját. Az RFD-k szemantikáját az L reguláris nyelv valamely előfordulásán értelmezzük. Ha csak egy RFD-ről van szó, a szemantika fogalma egyértelmű: egy adott előfordulás vagy kielégíti az adott RFD-t vagy nem. Ha viszont az RFD-knek egy Σ halmazáról akarunk valamit megállapítani (például logikai implikációt), megköveteljük az (1) Feltétel teljesülését, azt, hogy az összes Σ-beli függőséget valamely $I \in \mathcal{I}(L)$ (egyazon) előfordulás magában foglalja. Legyen $X \to Y \in \Sigma$, legyen $w = schema(I)$, akkor (1)-ből következik, hogy $nodes(w[X]) = nodes(X)$ és $nodes(w[Y]) = nodes(Y)$. Azaz, az I előfordulás sortípus duális mondata valamennyi szereplő kijelölést teljesen lefedi. Az (1) Feltétel teljesülése egyszerűsíti a logikai implikáció kezelését, elhagyásának következményeivel a 10.1. Fejezetben foglalkozunk.

6.3. Észrevétel. *Ha X egy kijelölés és $X \sqsubseteq I$ (legyen $w = schema(I)$), akkor $nodes(w[X]) = nodes(X)$.*

6.4. Észrevétel. *Ha X egy kijelölés és $X \sqsubseteq I$ (legyen $w = schema(I)$), akkor $nodes(w[X]) \neq \{\}$.*



10. ábra. Egy relációs sémának megfelelő véges automata gráfja

6.5 Definíció
Legyen L reguláris nyelv és legyen $M(L)$ az L nyelvet elfogadó véges automata gráf reprezentációja. Legyen Σ RFD-k egy halmaza és legyen $X \to Y$ egy RFD $M(L)$ felett, azt mondjuk, hogy Σ implikálja $X \to Y$-t (jelölve $\Sigma \models X \to Y$) ha minden olyan(véges) $I \in \mathcal{I}(L)$ adatbázis előfordulás esetén amely kielégíti Σ-t (feltételezve, hogy $(\Sigma \cup X \to Y) \sqsubseteq I$), $I \models X \to Y$ is teljesül.

6.1. Algoritmus. *Algoritmus RFD-k implikációjának ellenőrzésére.*
Input: az $M(L) = (V, E)$ gráf, az RFD-k egy Σ halmaza és a $\sigma : X \to Y$ RFD (ahol $X = (X_1, X_2)$ és $Y = (Y_1, Y_2)$) és mindegyik reguláris funkcionális függőség $M(L)$ felett értendő
Output: igaz, ha $\Sigma \models \sigma$, egyébként hamis

1. Inicializálás
TC=(V',E'):= $M(L)$ tranzitív lezártja
minden $e \in E$ color(e):= fekete
minden $v \in V$ color(v):= fekete
minden e' $\in$ E' color(e'):= kék
minden v' $\in$ V' color(v'):= kék
minden $v_X \in X_1$-re válasszuk $v(=v_X)$-t $\in V$ és legyen color(v):= zöld
minden $e_X \in E(X_2) \cap E$-re válasszuk $e(=e_X)$-t $\in E$ és legyen color(e):= zöld
minden $v_X \in X_1$ válasszuk v'$(=v_X)$-t $\in$ V' és legyen color(v'):= zöld
minden $e_X \in E(X_2)$-re válasszuk e'$(=e_X)$-t $\in$ E' és legyen color(e'):= zöld
minden $v_Y \in Y_1$ $(v_Y \notin X_1)$-re válasszuk $v(=v_Y)$-t $\in V$ és legyen color(v):= sárga
minden $e_Y \in E(Y_2) \cap E(e_Y \notin E(X_2))$-re válasszuk $e(=e_Y)$-t $\in E$ és legyen

color(e):= sárga
minden $v_Y \in Y_1\,(v_Y \notin X_1)$*-re válasszuk v'(=*$v_Y$*)-t* $\in$ *V' és legyen color(v'):= piros*
minden $e_Y \in E\,(Y_2)\,(e_Y \notin E\,(X_2))$*-re válasszuk e'(=*$e_Y$*)-t* $\in$ *E' és legyen color(e'):= piros*
2. $FDSET := \Sigma$*;*
3. $greene := X$ *(azt jelenti, hogy* $greene_1 := X_1$ *és* $greene_2 := X_2$*);*
4. ismételjük, amíg van alkalmazható függőség:
ha $W = (W_1, W_2) \to Z = (Z_1, Z_2) \in \Sigma$ *és* $W \subseteq greene$ *(abban az értelemben, hogy* $W_1 \subseteq greene_1$ *és* $W_2 \subseteq greene_2$*), akkor*
i. $FDSET := FDSET - (W \to Z)$*;*
ii. $greene := greene \cup Z$ *(azt jelenti, hogy* $greene_1 := greene_1 \cup Z_1$ *és* $greene_2 := greene_2 \cup Z_2$*);*
iii. minden $v_Z \in Z_1$*-re válasszuk* $v\,(= v_Z)$*-t* $\in V$ *és legyen color(v):= zöld*
iv. minden $e_Z \in E\,(Z_2) \cap E$*-re válasszuk* $e\,(= e_Z)$*-t* $\in E$ *legyen color(e):= zöld*
v. minden $v_Z \in Z_1$*-re válasszuk* $v'\,(= v_Z)$*-t* $\in V'$ *és legyen color(v'):= zöld*
vi. minden $e_Z \in E\,(Z_2)$*-re válasszuk* $e'\,(= e_Z)$*-t* $\in E'$ *és legyen color(e'):= zöld*

5. ha count_ nodes(V,sárga) = count_ nodes(V',piros) = 0 és count_ edges(E,sárga) = count_ edges(E',piros) = 0, akkor output=igaz, egyébként output=hamis.

6.1 Lemma
A 6.1. Algoritmus eredménye nem függ a Σ *és* $X \to Y$ *esetén használt (azonos) kiválasztási módszertől.*

Bizonyítás
A (4.5. - 4.6. és 6.1. - 6.2. Definíciók szerint használt) kiválasztási módszer duális mondatok részsorozatainak $M\,(L)$ *tranzitív lezártja kijelölt éleinek és csúcsainak bejárásán alapuló kiválasztására van hatással. A 6.1. Algoritmus az* $M\,(L)$ *tranzitív lezártja csúcsait és éleit kezeli, nem foglalkozik azok bejárásaiból keletkező duális mondatokkal, és figyelemmel a 6.3. Megjegyzésre, valamennyi résztvevő függőség ugyanazt a kiválasztási módszert alkalmazza, tehát a 6.1. Algoritmus lépéseit a kiválasztási módszer nem befolyásolja.*

6.1 Propozíció (Reguláris funkcionális függőségek implikációja)
Legyen L reguláris nyelv és legyen Σ *RFD-k egy halmaza és legyen* $X \to Y$ *egy RFD (valamennyien* $M\,(L)$ *felett), akkor* $\Sigma \models X \to Y$ *akkor és csak akkor, ha a 6.1. Algoritmus az* $M\,(L)$*,* Σ *és* $X \to Y$ *inputtal igaz eredményt hoz. Az implikáció teljesülése nem függ a* Σ *és* $X \to Y$ *esetén alkalmazott (közös) kiválasztási módszertől.*

Bizonyítás
Az algoritmus 1. lépése létrehoz egy előfordulást amely az $X \to Y$ függőséget megsérti (a zöld szín azt jelenti, hogy a két „sor" azonos X attribútumaiban, de a sárga és piros színek Y attribútumaiban eltérő értékeket jeleznek. A 4. lépés iteratíven alkalmazza Σ függőségeit, arra kényszerítve az előfordulást, hogy elégítse ki az FDSET-ből választott függőségeket. Ha a $W \subseteq greene$ feltétel egy $W \to Z \in FDSET$ függőségre nem teljesül valamelyik lépésben, akkor őt nem kell figyelembe vennünk a ciklus ezen lépésénél, mivel az előfordulás aktuális állapota triviálisan kielégíti ezt a függőséget (W azon csúcsai, amelyek nincsenek greene-ben vagy sárga vagy fekete színűek $M(L)$-ben és ők vagy piros vagy kék színűek a TC tranzitív lezárton, azaz, az aktuális „sorok" a $W \to Z$ bal oldalán nem egyeznek meg). Ha az 5. lépés igaz értékkel fejeződik be, az azt jelenti, hogy Σ valamennyi függőségének kielégítése az $X \to Y$ kielégítéséhez vezet, azaz, a függőségek implikációja teljesül. Ha az 5. lépés hamis eredményre vezet, azt jelenti, hogy van egy előfordulás, amely Σ valamennyi függőségét kielégíti de nem elégíti ki $X \to Y$, azaz, a $\Sigma \models X \to Y$ nem teljesül. Indirekt módon bizonyítjuk, hogy az algoritmus eredménye nem függ az iterációban alkalmazott függőségek sorrendjétől, azaz, az algoritmus végére keletkező greene gráf az $M(L)$, Σ és $X \to Y$ egy adott értékére egyértelmű. Tegyük fel, hogy a 6.1. Algoritmus két különböző, mondjuk A és B végrehajtása két különböző $greene_A \neq greene_B$ gráfot eredményezett és legyen $W^{(0)} \to Z^{(0)} \in \Sigma$ úgy, hogy $W^{(0)} \subseteq greene_A$ de $W^{(0)} \not\subseteq greene_B$. Akkor van egy $W^{(1)} \to Z^{(1)} \in \Sigma$ úgy, hogy őt az A folyamatban $W^{(0)} \to Z^{(0)}$ előtt kellett alkalmazni de rá a B folyamatban nem került sor. Így eljutunk egy $W^{(n)} \to Z^{(n)} \in \Sigma$ függőséghez úgy, hogy $W^{(n)} \subseteq X$ és $W^{(n)} \not\subseteq greene_B$, ellentmondás.

A propozíció utolsó mondata a 6.1. lemmából következik.

A következő, a relációs adatbázisok világából származó lemma, [1], reguláris funkcionális függőségekre is érvényes.

6.2 Lemma
Legyen Σ RFD-k egy halmaza és $X \to Y$ egy RFD ugyanazon $M(L)$ felett. Akkor $\Sigma \models X \to Y$ akkor és csak akkor, ha $Y \subseteq greene$ amikor a 6.1. Algoritmus véget ér.

Bizonyítás
A 6.1. Algoritmus konstrukciója biztosítja, hogy befejeződésekor a greene kijelölés az $M(L)$ gráf és tranzitív lezártja valamennyi zöld csúcsát és élét tartalmazza. Ha a 6.1. Algoritmus igaz értékkel tér vissza az $(\Sigma, X \to Y, M(L))$ inputtal, akkor Y valamennyi éle és csúcsa zöld lett, azaz, $Y \subseteq greene$. Egyébként, ha az algoritmus hamis eredményre vezet, akkor van(nak) csúcsa(i) és/vagy éle(i) Y-nak $M(L)$-ben és/vagy a tranzitív lezártjában amely(ek)nek valamely más színe van, azaz, $Y \not\subseteq greene$.

6.5 Megjegyzés
A 6.1. Algoritmus $O(|\Sigma|^2)$ időben fut, ahol $|\Sigma|$ az RFD-k száma Σ-ban. A relációs modellben a logikai implikáció lineáris időben eldönthető, mivel a relációs implikációs probléma az ítéletlogikai HORN-SAT problémával ekvivalens [17]. Egy $X \to Y$ reguláris FD és w *duális mondat esetén, legyen* $\mathrm{w}[X] = \mathrm{w}_1 \ldots \mathrm{w}_n$ *és legyen* $\mathrm{w}[Y] = z$*, akkor a megfelelő $W_1 \ldots W_n$ és Z ítéletlogikai változók száma* w*-től függ, azaz, nem korlátos ha a reguláris nyelv rekurzív.*

Másrészről, társíthatunk ítéletlogikai változókat az $M(L)$ gráf csúcsaihoz atomi kijelöléseket használva: legyen v az $M(L)$ egy csúcsa, akkor a v-hez rendelt atomi kijelölés legyen $X_v = (\{v\}, (\{\}, \{\}))$. Rendelhetünk igazságértékeket az atomi kijelölésekhez az RFD-k kiértékeléséhez hasonló módon: az X_v atomi kijelölés értéke legyen igaz (két sort, mondjuk $t_1, t_2; type(t_i) =$ w *bázisul választva), akkor és csak akkor, ha $t_1[X_v] = t_2[X_v]$.*

Definiálhatunk reguláris Bool-függőségeket az RFD fogalmának kiterjesztésével: az X ítéletlogikai változó, amelyet a (atomi vagy összetett) $X=(X_1, X_2)$ kijelöléshez társítunk az előbb mondottak szerint igazságértéket kaphat. A 6.1. Algoritmus nem tud egy reguláris Bool-függőséget eldönteni, mert az algoritmus az RFD által képviselt Bool-implikáción alapszik (ha a bal oldalt már kiszíneztük, akkor a jobb oldalt is ki kell színeznünk).

6.3. Reguláris funkcionális függőségek axiomatizálása

Az XML funkcionális függőségekre az általános esetben nem létezik véges axiomatizáció: Arenas és Libkin kimutatták [6] (egy, a k-méretű axiomatizálásra vonatkozó tétel alapján [1]) hogy a logikai implikáció az általuk leírt XML funkcionális függőségi osztály (tree tuples modell) esetében végesen nem axiomatizálható ha a sémaleírás (DTD) tetszőleges számú diszjunkciót tartalmazhat. Úgy látszik azonban , hogy a reguláris funkcionális függőségek esetében van egy helyes és teljes Armstrong-típusú axiómarendszer. Ez azért lehet így, mert az RFD nem igazából XML funkcionális függőség: az RFD mindössze egy „horizontális" dimenzióra korlátozódik, anélkül, hogy az XML fa-struktúráját modellezni tudná. Továbbá, az (1) feltétel kizárja a kezelhetetlen diszjunkciókat.

Felidézünk néhány hasznos jelölést:

Legyen L reguláris nyelv, legyen $M(L)$ a nyelvet elfogadó automata.

Legyen $X = (X_1, X_2)$ és $Y = (Y_1, Y_2)$ két kijelölés $M(L)$ felett, azt mondjuk, hogy Y tartalmazza X-et (jelölve $X \subseteq Y$) akkor és csak akkor, ha $X_1 \subseteq Y_1$ és $X_2 \subseteq Y_2$. $X_2 \subseteq Y_2$ azt jelenti, hogy ha $X_2 = (E_X, V_X)$ és $Y_2 = (E_Y, V_Y)$, akkor $E_X \subseteq E_Y$ és $V_X \subseteq V_Y$.

Legyen $X = (X_1, X_2)$ és $Y = (Y_1, Y_2)$ két kijelölés $M(L)$ felett. Az X és Y uniója (jelölve $Z = X \cup Y$) az a $Z = (Z_1, Z_2)$ kijelölés, amelyre $Z_1 = X_1 \cup Y_1$, $Z_2 = (E_Z, V_Z)$, ahol $E_Z = E_X \cup E_Y$ és $V_Z = V_X \cup V_Y$.

A reguláris funkcionális függőségre a három Armstrong axiómát a következőképpen adhatjuk meg:
Legyen L reguláris nyelv, legyen $M(L)$ a nyelvet elfogadó automata. Legyenek X, Y, Z kijelölések $M(L)$ felett.

1. Reflexivitás(*RFD-1*): ha $Y \subseteq X$, akkor $X \to Y$
2. Bővítés(*RFD-2*): ha $X \to Y$, akkor $X \cup Z \to Y \cup Z$
3. Tranzitivitás(*RFD-3*): ha $X \to Y$ és $Y \to Z$, akkor $X \to Z$

6.1. Tétel. *Az {RFD-1,RFD-2,RFD-3} axiómarendszer a reguláris funkcionális függőségek implikációjára nézve helyes és teljes.*

Bizonyítás
Ezen tétel kontextusában az (1) feltétel érvényességét adottnak vesszük.

Helyesség
Az axiómarendszer helyességének bizonyításához elég belátni, hogy az axiómák helyesek.

RFD-1: hivjuk fel a 6.1. Algoritmust $\Sigma = \emptyset$ és $\sigma : X \to Y$ inputtal. Az Inicializáció X valamennyi élét és csúcsát zöld színre színezi (egyidejűleg Y csúcsait és éleit is az azonos, zöld színre színezi, hiszen $Y \subseteq X$). Y végig zöld marad a további inicializációs lépések alatt, mert Y-nak csak azon csúcsait és éleit színeznénk piros és sárga színekre, amelyek nincsenek X-ben (ilyenek nincsenek, mert $Y \subseteq X$). Az algoritmus azonnal visszatér az igaz eredménnyel, mert rögtön az 5. lépésre ugrik és a kiértékelés azonnal az igaz értéket hozza ki. Ez azt jelenti, hogy az RFD-k egy üres halmaza implikálja $X \to Y$-t, azaz, minden előfordulás kielégíti azt.

RFD-2: hívjuk fel a 6.1. Algoritmust $\Sigma = \{X \to Y\}$ és $\sigma : X \cup Z \to Y \cup \cup Z$ inputtal. Az algoritmus alkalmazza $X \to Y$-t és „rendbe hozza" (zöld-re színezi) Y-t (Z már zöld, $X \cup Z$ miatt).

RFD-3: hívjuk fel a 6.1. Algoritmust $\Sigma = \{X \to Y, Y \to Z\}$ és $\sigma : X \to Z$ inputtal. Az algoritmus először alkalmazza $X \to Y$-t és rendbe hozza Y-t, azután alkalmazza $Y \to Z$-t és rendbe hozza Z-t.

Teljesség
Azt kell bizonyítanunk, hogy ha $\Sigma \models \sigma$, akkor van egy RFD-kből álló

$\sigma_0, \ldots, \sigma_n; n \geq 0$ *sorozat úgy, hogy* $\sigma_n = \sigma$, *és minden* $\sigma_i\,(0 \leq i \leq n)$*-re vagy* $\sigma_i \in \Sigma$, *vagy* σ_i *előállítható a* $\{\sigma_j, 0 \leq j < i\}$ *sorozatból a három axióma alkalmazásával (azt mondjuk, hogy* σ *bizonyítható* Σ*-ból).*
Legyen $\sigma = X \to Y$. *Hívjuk fel a 6.1. Algoritmust az* $(\Sigma, \sigma, M\,(L))$ *inputtal, ahol* $M\,(L)$ *az* L *reguláris nyelv elfogadó automatája (gráf). Feltesszük, hogy minden szóbajövő RFD* $M\,(L)$ *felett értelmezett, továbbá, hogy az (1) feltétel teljesül. A bizonyítás a relációs modell esetéhez hasonlóan vezethető le [1].*
Tegyük fel, hogy a 6.1. Algoritmus m *iterációs lépés után befejeződik* $(m \geq 0)$. *Jelölje* $greene_i$ *a greene kijelölés állapotát az i-edik iteráció után* $(greene_0 = X)$. m *szerinti indukcióval megmutatjuk, hogy létezik az* $X \to greene_m$ *bizonyítása* Σ*-ból. Ha* $m = 0$ *akkor* $\sigma_0 = X \to greene_0$ *a bizonyítás az RFD-1 axióma miatt. Az* r_i *indukciós lépésre kapjuk*
$\sigma_{r_i} = X \to greene_i$ *(indukciós feltevés)*
Hogy a 6.1. Algoritmus $greene_i$*-ből* $greene_{i+1}$*-hez jusson, kell, hogy legyen egy RFD* Σ*-ban úgy, hogy a bal oldala benne van* $greene_i$*-ben. Legyen* $W \to Z$ *ez az RFD, így* $W \subseteq greene_i$. *Akkor*
$\sigma_{r_i+1} = W \to Z\,(\in \Sigma)$
$\sigma_{r_i+2} = greene_i \to W$ *(RFD-1)*
$\sigma_{r_i+3} = greene_i \to Z$ *(RFD-3)*
A 6.1. Algoritmus ii lépése szerint $greene_{i+1} = greene_i \cup Z$, *akkor* $greene_i \to greene_i \cup Z$ *(RFD-2), és innen*
$\sigma_{r_i+4} = greene_i \to greene_{i+1}$ *(RFD-2)*
$\sigma_{r_i+5} = X \to greene_{i+1}$ *(RFD-3)*

Így kapunk egy bizonyítását $X \to greene_m$*-nek* Σ*-ból, azaz, van egy* $N \geq 0$ *úgy, hogy* $\sigma_N = X \to greene_m$. *Mivel* $\Sigma \models X \to Y$, *a 6.2. lemmából következik, hogy* $Y \subseteq greene_m$, *alkalmazva RFD-1-et kapjuk, hogy* $\sigma_{N+1} = greene_m \to Y$ *és RFD-3-mal, hogy* $\sigma_{N+2} = X \to Y$.

7. A reguláris FD értelmezése XML sémanyelveken

Ebben a fejezetben a reguláris funkcionális függőség definícióját alkalmazzuk XML sémanyelvekre.

A 2. Fejezetben láttuk, hogy egy XML dokumentum elemei egy reguláris nyelv mondatainak tekinthetők. Ehhez az (adat)nyelvhez társítható duális nyelvként a dokumentumot leíró DTD vagy XML-séma elemdeklarációból származó reguláris kifejezések által generált nyelv. Ahhoz, hogy a reguláris funkcionális függőséget értelmezhessük az XML környezetben vagy egy reguláris grammatikát kell megfeleltetnünk az XML dokumentumnak, vagy meg kell szerkesztenünk az előbb említett reguláris kifejezés(ek)hez tartozó véges automatát.

A megfeleltetést DTD elemleírás és XML-séma összetett típus példán szemléltetjük.

7.1. Reguláris FD DTD elemleírásból készített véges automatán

7.1. Példa. *A 11. ábrán látható dokumentum a 12. ábrán mutatott DTD-nek felel meg Ez a DTD tartalmazza a következő elemleírást:*

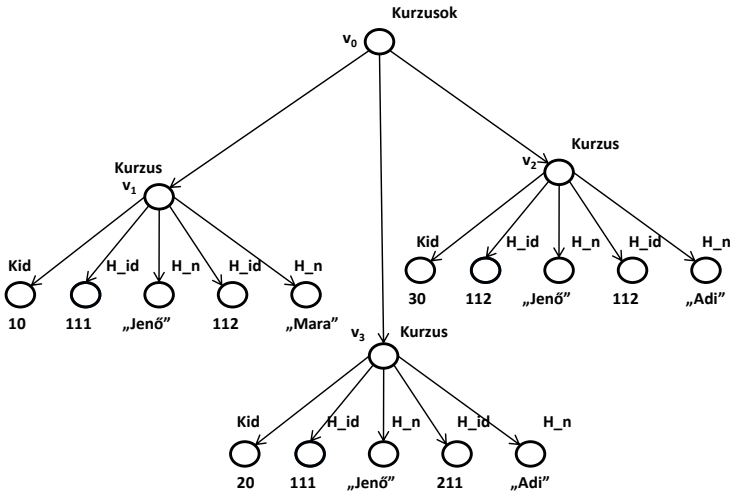
```
<!ELEMENT kurzus (Kid,(H_id,H_n)+)>
```

Ez a dokumentum egyetemi kurzusok néhány jellemző adatát (kurzusazonosító - Kid, és a kurzust felvevő hallgatók adatai: hallgató azonosítója - H_id; hallgató neve - H_n) adja meg. A kurzusazonosító elvárhatóan egyértelműen meghatározza a kurzus adatait (a példában a kurzust felvett hallgatók sorát). Ezt a megszorítást kulcs függőséggel írhatjuk le. A 13. ábra mutatja az elemleírásból, mint reguláris kifejezésből a Berry-Sethi algoritmussal (3. Fejezet) konstruált véges automatát.

Ezen az automatán megadhatjuk az előbb leírt kulcs függőséget:

$$RFD_{K1} : (\{Kid\}, (\{\}, \{\})) \rightarrow (\{\}, (\{H\_id, H\_n\}, \{(H\_id, H\_n)\}))$$

Informálisan fennáll, hogy a hallgatói azonosító egyértelműen meghatározza a hallgató nevét, azaz, van egy kulcs függőség, amely egy hallgató adatait (H_id,H_n) összekapcsolja:



11. ábra. Kurzus adatok XML dokumentuma

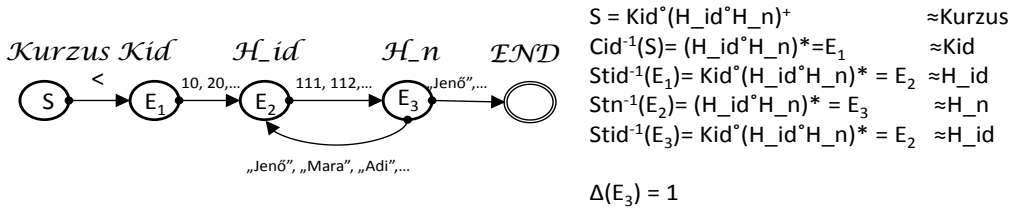
```
<!DOCTYPE Kurzusok [
<!ELEMENT Kurzusok (Kurzus)+>
<!ELEMENT Kurzus (Kid,(H_id,H_n)+)>
<!ELEMENT Kid (#PCDATA)>
<!ELEMENT H_id (#PCDATA)>
<!ELEMENT H_n (#PCDATA)>
]>
```

12. ábra. A 11. ábra - Kurzus adatok - XML dokumentumát leíró DTD

$RFD_{K2} : (\{\}, (\{H\_id\}, \{\})) \rightarrow (\{\}, (\{H\_n\}, \{\}))$

felhasználva a 13. ábrán látható automatát és a 6. Fejezetben adott definíciókat. Látni kell azonban, hogy a 4.5 - 4.6 definíciók szerinti kiválasztási módszerek egyike sem fejezi ki a kívánt megszorítást (H_ id a „testvér” H_ n kulcsa), így ennek a kezelésére egy új kiválasztási módszert kell alkalmaznunk, egy olyan módszert, amely együtt ismételt csoportokból képes választani. Erre az esetre a 7.1-7.2 (indexelt) definíciók alkalmazhatók.

7.1 Definíció (Indexelt kiválasztás)
Legyen L reguláris nyelv és legyen $M(L)$ az L nyelvet elfogadó véges automata gráf reprezentációja és legyen w *duális mondat $M(L)$ felett. Legyen*



13. ábra. A Kurzus/Hallgatók dokumentum elfogadó automatája

$c = (c_1, \ldots, c_{n+1}), n \geq 1$ *ciklus (kör)* $walk\,(\mathrm{w})$*-ben úgy, hogy* $c_{n+1} = c_1$ *és* $c_i \neq$ $= c_j, 1 \leq i < j \leq n$. *Legyen* $Y = (Y_1, Y_2)$ *egy (kiválasztandó csúcsokat és éleket leíró) kijelölés* $M\,(L)$ *felett úgy, hogy* $nodes\,(Y_2) = \{y_1, \ldots, y_m\}, m \leq n-1$ *a* c *csúcsainak egy halmaza (minden* $y_k, 1 \leq k \leq m$ *esetén van egy* $1 \leq j \leq n$ *úgy, hogy* $y_k = c_j$*). Legyen* $i \geq 1$ *egy index érték. A* $nodes\,(Y_2)$ *csúcsokat az* i *indexre kiválasztjuk, jelölve* $\mathrm{w}\left[Y^{(i)}\right]$ *amikor a bejárás érinti őket a* c *ciklus* i*-edik záródásakor.*

7.2 Definíció (Indexelt reguláris funkcionális függőség)
Legyen L *reguláris nyelv és legyen* $M\,(L)$ *az* L *nyelvet elfogadó véges automata gráf reprezentációja és legyen* w *duális mondat* $M\,(L)$ *felett. Legyenek* X *és* Y *az* $M\,(L)$ *felett, a 7.1. Definíció szerinti kiválasztási módszert alkalmazó kijelölések. Egy indexelt reguláris funkcionális függőség* (RFD^{ind}) $M\,(L)$ *felett a* $X \xrightarrow{ind} Y$ *formájú kifejezés. Az* $I_{\mathrm{w}} \in \mathcal{I}\,(L)$ *(véges) adatbázis előfordulás kielégíti a* $X \xrightarrow{ind} Y$ *indexelt funkcionális függőséget (jelölve* $I_{\mathrm{w}} \models X \xrightarrow{ind} Y$*), ha bármely két* $i, j \geq 1$ *index értékre és bármely két* $t_1, t_2 \in I_{\mathrm{w}}$ *sorra ha* $t_1\left[X^{(i)}\right] =$ $= t_2\left[X^{(j)}\right]$ *teljesül, akkor* $t_1\left[Y^{(i)}\right] = t_2\left[Y^{(j)}\right]$ *is igaz.*

7.2. Példa. *A 11. ábra XML dokumentumához rendelt duális mondat*

$\mathrm{w} = \lfloor Kid \mid H\_id \mid H\_n \mid H\_id \mid H\_n \rceil$.

A dokumentum 3, azonos típusú (type $(t_i) = \mathrm{w}, 1 \leq i \leq 3$) sort tartalmaz, nevezetesen:

$t_1 = \lfloor 10 \mid 111 \mid Elek \mid 112 \mid Mara \rceil$
$t_2 = \lfloor 30 \mid 112 \mid Elek \mid 112 \mid Adi \rceil$
$t_3 = \lfloor 20 \mid 111 \mid Elek \mid 211 \mid Adi \rceil$

Idézzük fel az RFD_{K2}:$X \rightarrow Y$ függőséget, amelyben $X = (\{\}, (\{H\_id\}, \{\}))$, $Y = (\{\}, (\{H\_n\}, \{\}))$ a 7.1. Példa szerint.
A szigorú kiválasztással $\mathrm{w}[X] = \lfloor H\_id \mid H\_id \rceil$, $\mathrm{w}[Y] = \lfloor H\_n \mid H\_n \rceil$ jelsorozatokhoz jutunk. A fenti sorokból ezek a jelsorozatok az alábbi értékeket vetítik ki:

$t_1[X] = \lfloor 111 \mid 112 \rceil \quad t_1[Y] = \lfloor Elek \mid Mara \rceil$
$t_2[X] = \lfloor 112 \mid 112 \rceil \quad t_2[Y] = \lfloor Elek \mid Adi \rceil$
$t_3[X] = \lfloor 111 \mid 211 \rceil \quad t_3[Y] = \lfloor Elek \mid Adi \rceil$

A 11. ábra XML dokumentuma nyilvánvalóan kielégíti a RFD_{K2} függőséget: a bal oldalak rendre különböznek. Azaz, RFD_{K2} nem képes felfedni a hallgatói azonosító és a hallgató neve közötti informális kulcsösszefüggés sérülését: A H_id azonosító 112 értékéhez különböző H_n értékek (Elek, Mara) tartoznak.
Fogalmazzuk újra RFD_{K2}-t a 7.1,7.2 definíciók segítségével:
$RFD^{ind}_{K2}: X \xrightarrow{ind} Y$, ahol $X = (\{\}, (\{H\_id\}, \{\}))$, $Y = (\{\}, (\{H\_n\}, \{\}))$, így az alábbi jelsorozatokat kapjuk:

$t_1\left[X^{(1)}\right] = 111 \quad t_1\left[X^{(2)}\right] = 112$
$t_1\left[Y^{(1)}\right] = Elek \quad t_1\left[Y^{(2)}\right] = Mara$
$t_2\left[X^{(1)}\right] = 112 \quad t_2\left[X^{(2)}\right] = 112$
$t_2\left[Y^{(1)}\right] = Elek \quad t_2\left[Y^{(2)}\right] = Adi$
$t_3\left[X^{(1)}\right] = 111 \quad t_3\left[X^{(2)}\right] = 211$
$t_3\left[Y^{(1)}\right] = Elek \quad t_3\left[Y^{(2)}\right] = Adi$

Ez a RFD^{ind}_{K2} függőség már észleli a megszorítás megsértését: $t_1\left[X^{(2)}\right] = 112 = t_2\left[X^{(1)}\right]$, de $t_1\left[Y^{(2)}\right] = Mara \neq t_2\left[Y^{(1)}\right] = Elek$

7.2. Reguláris FD XML-séma összetett típusából vett grammatikán

Az XML-séma W3C specifikációja [47] tartalmaz épségi megszorításokat (min és max occurs kifejezések, azonossági megszorítások amelyek felölelik az egyértelműségi, kulcs és idegen kulcs definíciókat). A reguláris funkcionális függőség koncepciója az XSD összetett típusára alkalmazva, azaz, egy RFD definiálása valamely összetett típus mint reguláris kifejezés által generált reguláris nyelven, elhelyezhető az XML séma azonossági megszorításainak hierarchiájába. A reguláris funkcionális függőséggel kezelhetőek a *min/maxOccurs* kardinalitási megszorítások, a *sequence* és *choice* jelzők (bár az *all* indikátor realizálása kissé körülményes lehet, valójában *valamennyi* utat figyelembe kellene vennünk). Az RFD aktuális modellje az *XSD Attributes* és *XSD Mixed Type* attribútumokat nem tudja kezelni.

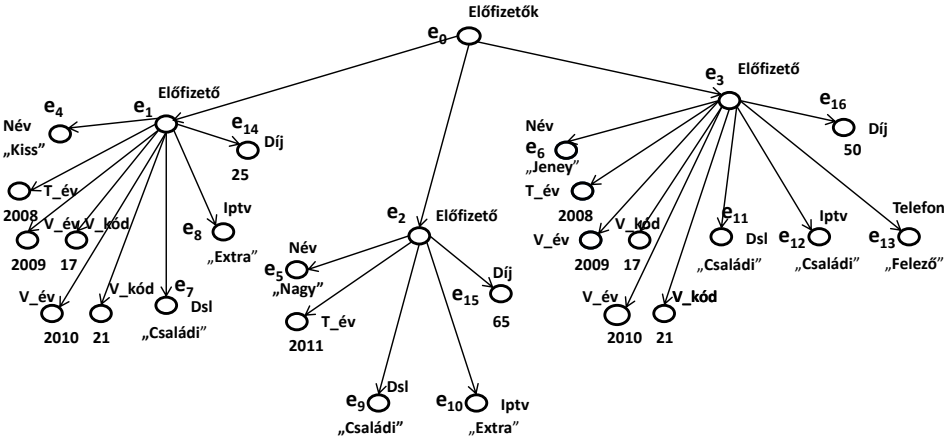
Egy XML dokumentumhoz a következőképpen társíthatunk egy reguláris grammatikát:

7.3 Definíció (XML elem grammatikája)
Legyen $G = (N, T, S, P)$ *reguláris grammatika és legyen* D *egy XML dokumentum. Legyen ele(e) az a függvény, amely a* T *XML fa* e *csúcsának az elemnevét adja meg. Legyen* E_S *a* T *azon csúcsainak halmaza, amelyek elemneve éppen az* S *nemterminális szimbólum, azaz,* $E_S = \{e | e \in T, ele(e) = S\}$. *Azt mondjuk, hogy* G *társítható* T*-vel, ha van egy* $A : E_S \mapsto L(G)$ *leképezés, úgy, hogy ha* $e \in E_S$ *akkor* $A(e) = a_1 a_2 \dots a_m \in L(G)$ *és az alábbiak teljesülnek: Az* e *csúcsnak pontosan* m *gyermeke van; legyenek ezek* $e_1, e_2, \dots, e_m$*, akkor a helyettesítési szabályoknak egy láncolata*

$\{S \Rightarrow \langle N_1, \dots, N_{m-1} \Rightarrow a_{m-1} N_m, N_m \Rightarrow a_m \mathcal{ENTD}, \mathcal{ENTD} \Rightarrow \rangle\}$ *(ahol* $\langle$ *megfelel* S *nyitó és* $\rangle$ *a záró jelölőjének)*

kell, hogy létezzen P*-ben úgy, hogy a nemterminális szimbólumok* $N_1, N_2, \dots, N_m$ *tömbje megegyezik az* $ele(e_1), \dots, ele(e_m)$ *elemnevek tömbjével, és az* e_i *csúcsok szöveg értékei (címkék) a* $a_i, 1 \leq i \leq m$ *terminális szimbólumokkal egyenlőek.*

Az A leképezéshez tartozik a $D : E_S \mapsto D(L(G))$ leképezés is, amely az XML fa csúcsait a duális nyelvbe képezi le. A fenti Definíció jelöléseit használva, $D(e) = N_1 N_2 \dots N_m$.



14. ábra. XML példa dokumentum telekom előfizetői adatokra

7.3. Példa. *Az XML dokumentum a 14. ábrán egy nyilvános telefonhálózat előfizetői adatbázisát mutatja. A 15. ábrán látható az előfizetői adatbázist leíró XSD sémaleírás, benne az Előfizető elem t_Előfizető összetett típusa. Ez a definíció lényegében az alábbi reguláris kifejezésnek felel meg:*
$E = \left(Név, T\_év, (V\_év, V\_kód)^*, Dsl, Iptv^?, Telefon^?, Díj\right)$

Legyen $G = (N, T, S, P)$ egy reguláris grammatika az alábbiak szerint:

$N = \{Előfizető, Név, T\_év, V\_év, V\_kód, Dsl, Iptv, Telefon, Díj\}$
$T = \{Kiss$,2008,17, 2009, $\ldots\}$
$S = Előfizető$
P=$\{Előfizető \Rightarrow Kiss\, T\_év, T\_év \Rightarrow 2008\, V\_év, \ldots\}$

A 2. táblázat a 14. ábra XML dokumentumának a G grammatikával való A társítását adja meg.

Ez a társítás az XML dokumentum Előfizető elemeket reprezentáló csúcsait az $L(G)$ nyelv egy részhalmazába képezi le.

Egy adott Előfizető-höz tartozó T_év elem a (telefonvonal)kapcsolat kiépítésének (aktiválásának) kezdő évét adja meg. Ettől az évtől kezdve a telefontársaság a vonalat évente ellenőrzi. Minden ellenőrzési eseményhez, a V_év elem tartalmazza az ellenőrzés év adatát, a V_kód elem pedig egy speciális kódot amely a T_év elem értékétől és az ellenőrzés évétől (V_év elem)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
               elementFormDefault="unqualified">
  <xs:element name="Előfizetők">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded" minOccurs="1">
        <xs:element name="Előfizető" type="t_Előfizető"
                 maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="t_Előfizető">
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:element name="Név" type="xs:NCName"/>
      <xs:element name="T_év" type="xs:integer"/>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="V_év" type="xs:integer"/>
      <xs:element name="V_kód" type="xs:integer"/>
  </xs:sequence>
      <xs:element name="Dsl" type="xs:NCName"/>
      <xs:element minOccurs="0" name="Iptv"
                                type="xs:NCName"/>
      <xs:element minOccurs="0" name="Telefon"
                                type="xs:NCName"/>
      <xs:element name="Díj" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

15. ábra. Az *Előfizetők* sémaleírása, benne az *Előfizető* elem (14. ábra) összetett típus definíciója

$D(e_1) =$	Név	T_év	V_év	V_kód	V_év	V_kód	Dsl	Iptv	Díj
$A(e_1) =$	Kiss	2008	2009	17	2010	21	Családi	Extra	25
$D(e_2) =$	Név	T_év	Dsl	Iptv	Díj				
$A(e_2) =$	Nagy	2011	Családi	Extra	65				
$D(e_3) =$	Név	T_év	V_év	V_kód	V_év	V_kód	Dsl	Iptv	Telefon Díj
$A(e_3) =$	Jeney	2008	2009	17	2010	21	Családi	Családi	Felező 50

2. táblázat. XML adatok leképezése reguláris nyelv mondataiba

3. táblázat. Nyilvános telefonhálózat előfizetői díjai

Szolgáltatás	Csomag díj							
DSL	Alap	10	Családi	20	Extra	25	Maximum	40
IPTV	Alap	10	Családi	15	Extra	25		
Telefon	Alap	5	Favorit	10	Felező	15		

függ. Fennáll tehát egy épségi megszorítás az adatok között, ezen megszorítást a fenti A társításon alapuló alábbi RFD-vel (tetszőleges kiválasztási módszert használva) írhatjuk le:

$$RFD_{1a} : (\{T\_év\}, (\{\}, \{\})) \rightarrow (\{\}, (\{V\_év, V\_kód\}, \{(V\_év, V\_kód)\}))$$

Az XML dokumentumban realizálódó adatbázis előfordulás kielégíti RFD_{1a}-t, mert két olyan sor van, a Kiss és Jeney előfizetők adatai, ahol a függőség bal oldala azonos értékeket tartalmaz (T_év=2008), és itt a jobb oldalak (2009 17 2010 21) is azonosak.

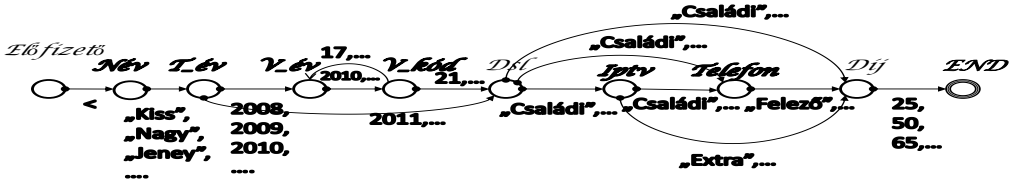
7.4. Példa. *Az XSD a 15. ábrán leírja egy nyilvános telefonhálózat előfizetőinek adattípusait és azok kapcsolatait. A 14. ábrán ezen XSD egy előfordulását, egy, az XSD-nek megfelelő XML dokumentumot láthatunk (legyen a neve T), a T dokumentum adatsorait mint mondatokat tartalmazó reguláris nyelvet elfogadó véges automatát a 16. ábra mutatja. Egy adott előfizető számára nyújtott szolgáltatások nem szükségképpen teljeskörűek, a Telefon vagy Iptv szolgáltatás igénybe vétele opcionális: lehetséges Dsl-t használni akár (vonalas) Telefon kapcsolat nélkül is, használhatunk Iptv-t vagy eltekinthetünk attól. A Díj elem mutatja az igénybe vett szolgáltatások havonta fizetendő alapdíját.*

Ez a díj egyedül a nyújtott szolgáltatásoktól függ a 3. táblázat szerint (például, Telefon: Alap=5,Favorit=10,Felező=15 stb.), azaz, ha két előfizető azonos szolgáltatásokat rendelt meg akkor a havidíjuk azonos lesz. Így egy érvényes adatbázis ki kell, hogy elégítse az igénybe vett szolgáltatások és a havonta fizetendő díj között fennálló reguláris funkcionális függőséget:
$RFD_{1b} : (\{Dsl, Iptv, Telefon\}, (\{\}, \{\})) \rightarrow (\{Díj\}, (\{\}, \{\}))$

RFD_{1b} a T három duális mondatából két különböző részsorozat-párt választ ki:

$\mathrm{w}_1 = \lfloor Név \mid T\_év \mid V\_év \mid V\_kód \mid V\_év \mid V\_kód \mid Dsl \mid Iptv \mid Díj \rceil$
$\mathrm{w}_1[X] = \lfloor Dsl \mid Iptv \rceil, \mathrm{w}_1[Y] = Díj$
$\mathrm{w}_2 = \lfloor Név \mid T\_év \mid Dsl \mid Iptv\, Díj \rceil$
$\mathrm{w}_2[X] = \lfloor Dsl \mid Iptv \rceil, \mathrm{w}_2[Y] = Díj$
$\mathrm{w}_3 =$
$= \lfloor Név \mid T\_év \mid V\_év \mid V\_kód \mid V\_év \mid V\_kód \mid Dsl \mid Iptv \mid Telefon \mid Díj \rceil$
$\mathrm{w}_3[X] = \lfloor Dsl \mid Iptv \mid Telefon \rceil, \mathrm{w}_3[Y] = Díj$

Mivel a duális mondatok különbözőek, T a kapcsolódó kiterjesztett reláció három előfordulását tartalmazza, ezek mindegyike triviálisan kielégíti RFD_{1b}-t. Tágabban értelmezve az előfordulás fogalmát (reguláris mondatok véges halmaza, részletesebben a 10.2. Fejezetben) az RFD_{1b} érdemi megszorítást fejez ki: $\mathrm{w}_1[X] = Dsl\, Iptv = \mathrm{w}_2[X]$ *és* $\mathrm{w}_1[Y] = Díj = \mathrm{w}_2[Y]$*, az erős kiértékeléssel (6.2. Megjegyzés), a Kiss és Nagy előfizetők esetén $t_i[X] = Családi\, Extra; i = 1,2$ egyenlő bal oldalakat kapunk, ám a jobb oldalak különböznek: $t_1[Y] = 25 \neq 65 = t_2[Y]$, így az XML dokumentum mint reguláris nyelv az RFD_{1b} függőséget erősen nem elégíti ki ($T \overset{s}{\not\models} RFD_{1b}$), azaz, RFD_{1b} erős kiértékeléssel felismeri a szolgáltatás/díjfizetés kapcsolatra vonatkozó épségi megszorítás megsértését. (A gyenge kielégítés viszont teljesül,$T \overset{w}{\models} RFD_{1b}$, mivel mindegyik sor különböző típusú).*



16. ábra. Hiányos (opcionális) adatokat leíró XML elemtípus FSA gráfja

8. A reguláris FD helye a nem-relációs FD-k között

Ebben a Fejezetben az RFD koncepcióját vetjük össze a nem-relációs funkcionális függőség (XFD, összetett értékű FD) fogalmakkal, elsősorban az értelmezési tartományok és a (többnyire összetett) értékek alkalmazott összehasonlítási eljárásai alapján. Az RFD általában abban tér el az ismertetett eljárásoktól, hogy az RFD rendezett halmazokkal (reguláris mondatokkal) dolgozik. A 10.3. Fejezetben foglalkozunk az RFD rendezetlen szimbólumhalmazokra kiterjesztésével.

Valamennyi látókörünkbe került XML funkcionális függőség (XFD) koncepció foglalkozik reguláris kifejezésekkel és/vagy reguláris nyelvekkel. Arenas és Libkin az általuk bevezetett *tree tuple* XFD modellben, [6], az XFD-k logikai implikációjával kapcsolatban a kezelt reguláris kifejezés típusától függően különböző nehézségi fokú bonyolultsági eredményeket bizonyítanak. Kvadratikus idejű bonyolultságot bizonyítanak az *egyszerű*-nek nevezett reguláris kifejezés esetében (a mi modellünk ezzel az esettel vág egybe, ha a (1) Feltételt adottnak vesszük). Bebizonyítják, hogy az általános esetben a logikai implikáció végesen nem axiomatizálható. A 9. Fejezetben részletesen összehasonlítjuk a *tree tuple* modellt az RFD modellel.

Yu és Jagadish [58] azt találták, hogy a *tree tuple* modell nem tud kezelni olyan megszorításokat amelyeknél az adatelemeknek egy halmazát együtt, egységként kellene tekinteni. (A 9.1. Fejezet utolsó bekezdése a *tree tuple* XFD és RFD hasonló különbségére utal). Emiatt javasolják a *tree tuple* modellnek *set* (többszörös előfordulású elemek) komponensekkel való bővítését. A *set* komponensek összehasonlítására alkalmazott módszerük a miénktől eltér: az RFD rendezett *set* komponensekkel (szimbólum sorozatokkal) dolgozik. Az RFD és a kiterjesztett *tree tuple* modell összehasonlítása a 9.2. Fejezetben található.

Hartmann és Link [25, 32] (2003) az XFD-t egy XML fa (dokumentum) séma-gráfján definiálják. Modelljükben az XFD az XML séma-gráf egy csúcsához tartozó részgráf. Egy XML fa kielégít egy ilyen XFD-t ha bármely két maximális részmásolat esetén a függőség bal és jobb oldalainak megfelelő projekciók teljesítik az XFD által meghatározott feltételt. A séma-gráf egy részgráfjának részmásolata az XML fa egy vele izomorf részgráfja, az XML fa séma-gráfba való homomorf leképezése szerint. Erre az XFD-re vonatkozóan két értékelési módszert adnak meg: az első hasonló a *tree-tuple* koncepciónál használthoz, a második testvér elemek rendezetlen listáit veti össze. Ez a módszer ismét különbözik az RFD-nél alkalmazottól: az RFD rendezett listákat (szimbólum sorozatokat) kezel.

Kot és White [35] újrafogalmazzák a *tree-tuple*-alapú XFD-t, mindenesetre DTD nélkül: az XML fa sorait egy *fa sablon* illesztései jelenítik meg. Egy XML dokumentum *fa sablon*-jai (tree patterns) egy beágyazott relációs adatbázist alkotnak. A kibontott relációs adatbázisra alkalmazva a klasszikus Chase algoritmust Arenas és Libkin [6] fentebb idézet eredményéhez hasonló komplexitású implikációt ellenőrző eljárás készíthető reguláris kifejezések különböző osztályai esetére. Másrészt, Kot és White bizonyítani tudják a logikai implikáció véges axiomatizálhatóságát (a nem-diszjunktív esetben), az Atzeni és Morfuni [7] által a null értékeket megengedő relációs FD-kre bevezetett axiómákkal.

Wang és Topor [56] séma definíció nélküli XML fákon definiálnak XFD-t: XML elemek címkéiből épülő útkifejezéseket használnak. Így olyan megszorítások is kezelhetőek, amelyek a *tree tuple* vagy *closest node* modellel nem fejezhetőek ki (9.1. és 9.3. Fejezet). A szemantikai értékeléshez három különböző hasonlítási eljárást határoznak meg: csúcs, halmaz és metszési megegyezés (node, set, interdect). A halmaz-egyezés definíciója eltér az RFD módszerétől: ignorálják a rendezettséget, RFD viszont rendezett halmazokkal operál.

Liu és kollégái [38, 59] megszorításokat megőrző eljárásokat (XML nézet) hoznak létre relációs adatbázisok XML dokumentumként való kezeléséhez. Alkalmazott modelljük, beágyazott relációk értelmezése XML keretben sokban hasonlít a mi megközelítésünkhöz: mi a megszorításokat egy adott típuson értelmezzük, amely típus előfordulásai az XML dokumentumon szétszóródva jelennek meg.

Buneman és mukatársai [11, 12] reguláris ösvénykifejezésket használnak: bevezetik a reguláris ösvénykifejezések egy osztályát és XML kulcsokat, azaz, XML funkcionális függőségeket definiálnak ezen osztály segítségével az ösvénykifejezések nyelvén. A kulcsok a reguláris ösvénykifejezések nyelv mondatainak halmazai, azaz, a funkcionális függőséget reguláris útkifejezésekből, mint mondatokból képezik. Ez a koncepció alapvetően eltér a miénktől, mivel az RFD modellben a funkcionális függőség a reguláris nyelv mondatain hat. Buneman és társai az általuk definiált XML kulcs osztályra megadják a logikai implikációra vonatkozó levezetési szabályok egy helyes és teljes rendszerét.

Hartmann és Link [29] javítják Buneman és társai fenti eredményét, bizonyítva, hogy a kulcsok implikációjára vonatkozó levezetési szabályok helyessége nem teljeskörű; az XML kulcsok logikai implikációjára megadnak egy másik axiomatizációt és bizonyítják helyességét valamint teljességét is. Nu-

merikus megszorításokkal kibővítik a reguláris ösvénykifejezésekre alapozott XML kulcs modellt [30]: a kiterjesztett modell speciális esete, min=max=1 éppen az XML kulcs. A numerikus megszorítások egy szülő gyermekeinek a számára vonatkoznak (kardinalitás). A mi modellünk nem használ ciklusszámlálókat, bár az indexelt RFD (7.1. Fejezet) az elemek összehasonlításakor figyelembe veszi az elemek ciklusbeli pozícióját. A 10.4. Fejezetben az RFD-t numerikus megszorításokkal bővítjük.

Összetett értékű adatbázisokat a relációs modellben beágyazott relációk formájában értelmeztek [1]. Az RFD modell implicit beágyazást valósít meg amikor a generált szimbólumok egy sorozatát egységként kezeli.

Intuitíve, egy XML dokumentumot felfoghatunk beágyazott elemek, azaz, általánosított sorok halmazának. Fischer és szerzőtársai [20] kiterjesztették a relációs FD (és a többértékű függőség, MVD) fogalmát beágyazott relációs adatbázisokra. Azt találták, hogy a beágyazás (kibontás) felfogható mint funkcionális (többértékű) függőség: egy FD úgy szeletel föl egy relációt, hogy a bal oldal egy adott értékéhez tartozó partíció részhalmaza a megfelelő jobb oldali érték által meghatározott partíciónak. Másrészt, ha a sorok egy csoportját beágyazási célból össze akarjuk vonni, akkor szükségünk van egy horizontális dekompozícióra, azaz particionálásra.

Hara és Davidson [24] beágyazott relációs adatbázison definiálnak *beágyazott funkcionális függőséget* (*nested functional dependencies* (NFD)) (a szigorú modellt használják, azaz, a sor és halmaz konstruktorok alternálnak). Bevezetnek attribútum-, sor- és halmazcímke komponensekből álló ösvénykifejezéseket és NFD-ket definiálnak ezeket a kifejezéseket használva (a jobb oldalon csak egyetlen ösvénykifejezést engednek meg). Ez a modell sokban hasonlít más XFD definíciókhoz. Hara és Davidson bemutatnak egy nyolc szabályból álló helyes és teljes levezetési rendszert az NFD-k logikai implikációjára.

Hartmann, Link és Schewe [27] szintén beágyazott relációs adatbázisokon definiálnak funkcionális függőséget. Induktív módon specifikálják a beágyazott adatbázisok egy absztrakt modelljét. A modell a sor, halmaz, hatványhalmaz, többes-halmaz és lista konstruktorokra nézve zárt. A modell összetett értékű attribútumain funkcionális függőséget definiálnak. Az absztrakt modellben Armstrong-típusú axiómákat definiálnak. Bebizonyítják, hogy a logikai implikáció axiomatizálása ezekkel az axiómákkal helyes és teljes. A levezetési szabályok teljességét úgy bizonyítják, hogy az FD-k valamely halmazából le nem vezethető FD esetén felállítanak egy kételemű ellenpéldát. Egy másik cikkben [26] a beágyazott adatbázis modellt az XML-re alkalmazzák:

a dokumentum típus definícióit beágyazott attribútumokkal reprezentálják, rekord, lista és diszjunkciós konstruktorok felhasználásával.

Hartmann és Link [28] az előbb említett absztrakt modellt tetszőleges (beágyazott) Bool függőségekkel bővítették ki. Bool (ítéletlogikai) változókat társítottak a beágyazott adatbázis attribútumaihoz és igazságértékeket rendeltek ezekhez a változókhoz a szokásos két-sor módszerrel (X igaz akkor és csak akkor, ha $t_1[X] = t_2[X]$). Megmutatják, a beágyazott Bool függőségek logikai implikációjának bonyolultsága az ítéletlogikai feladatéval azonos: NP-teljes a legáltalánosabb esetben.

Sali és Schewe [46] összetett értékű adatbázis modellen definiáltak funkcionális függőséget sor, lista, halmaz és multihalmaz konstruktorokkal.Azt találták, hogy a (diszjunkt) unió konstruktor esetén az FD-k logikai implikációjának az axiomatizációja nem teljesen valósítható meg. Ezért az FD-ket a számlálómentes függőségek egy osztályára szűkítették és erre az osztályra találtak helyes és teljes véges axiomatizációt. Mivel az RFD számlálómentes, ez az eredmény jól egyezik az RFD modellben érvényessel.

9. A reguláris FD összehasonlítása néhány XFD-vel

Ebben a fejezetben a reguláris funkcionális függőség koncepcióját hasonlítjuk össze három, a szakirodalomban kiemelten hivatkozott XFD definícióval.

A 7. Fejezetben láttuk, hogy a Reguláris FD az XML világban jól interpretálható. Számos érdekes XML funkcionális függőség (XFD) definíciót publikáltak: Wang 2005-ben [55] felmért és összehasonlított egy sereg különböző XML funkcionális függőség definíciót és egy új XFD-t javasolt, amely egyesíti és általánosítja a vizsgált XFD-ket: ezen XFD definíciók mindegyike DTD-kből vagy XML-sémákból konstruált útkifejezéseken alapul. A fő különbség ezen XFD koncepciók és a reguláris funkcionális függőség között az, hogy az RFD csak egy elemtípus deklarációt tud kezelni: a szemlélete „horizontális" egy XML fa vonatkozásában. Ezzel szemben, az XFD definíciók alapjában „vertikális" közelítést használnak: a gyökérből induló útkifejezésekkel hivatkoznak az XFD hatókörére, további, a hatókörhöz képest relatív útkifejezésekkel adják meg a függőség komponenseit. A komponensek az XML fa különböző szintjein lévő elemekre is hivatkozhatnak.

9.1. A reguláris FD összehasonlítása a „tree tuple" XFD-vel

A *tree tuple*[1] koncepciót Arenas és Libkin fogalmazta meg [6]. Egy *tree tuple* egy DTD teljes fájának egy leképezése egy a DTD-nek megfelelő XML dokumentumba (XML fa). Így a fő különbség a *tree tuple* XFD és az RFD között („vertikális" a „horizontálissal" szemben) azonnal látszik. Mint már említettük a 6.3. Fejezetben, a logikai implikáció az RFD esetében (de nem így az XFD-nél) végesen axiomatizálható.

Most egy példán összehasonlítjuk Arenas és Libkin [6] *tree tuple* koncepcióját a reguláris FD-vel.

9.1. Példa. *A 17. ábrán mutatott DTD ugyanazokat az adatokat írja le mint amelyeket a 7.3. Példában vizsgáltunk (XML-sémaleírással); a DTD-nek megfelelő XML dokumentum a 14. ábrán látható. Az adatmodell sugall egy épségi megszorítást: a havonta fizetendő Díj az igénybe vett szolgáltatások díjainak összegével egyenlő kell, hogy legyen.*

Itt most nem adjuk meg a *tree tuple*, az XML fák közötti *magábanfoglalás*, a DTD-nek való *megfelelés* és *kompatibilitás* definícióit, és a $trees_D, tuples_D$

[1] nem adunk magyar nevet ennek az XFD-nek, mert magyarul furcsán festene

```
<!DOCTYPE Előfizetők [
<!ELEMENT Előfizetők (Előfizető*)>
<!ELEMENT Előfizető (Név,T_év,(V_év,V_kód)*,Dsl,Iptv?,Telefon?,Díj)>
<!ELEMENT Név (#PCDATA)>
...
<!ELEMENT Díj (#PCDATA)>
]>
```

17. ábra. A 14. ábra XML dokumentumának DTD leírása

kifejezések pontos definícióit (Ezek megtalálhatók Arenas és Libkin hivatkozott cikkében [6]). $tuples_D$ a *tree tuple*-ek azon halmaza amely egy T XML fát hoz létre és összekapcsolja azt a kiindulási XML fát leíró DTD-vel.

9.2. Példa. *A 9.1. Példa megfogalmaz egy épségi megszorítást a 14. ábra adataira: az Előfizető által havonta fizetendő Díj az adott Előfizető által igénybe vett szolgáltatások függvénye. A tree tuple koncepció alapján megfogalmazott XFD ezt a megszorítást a következő útkifejezésekkel írja le:*

S_1 *:Előfizetők.Előfizető.Dsl,*
Előfizetők.Előfizető.Iptv,
Előfizetők.Előfizető.Telefon $\rightarrow$ S_2 *: Előfizetők.Előfizető.Díj*

Ez az XFD felismeri a megszorítás megsértését ha az üres értékeket ($\perp$) *egyenlőnek tekintjük. A 18. ábrán megadtunk két tree tuple-t (a „Kiss" és „Nagy" csúcsokra illeszkedve), ezeken láthatjuk az XFD megsértését. Használva a p: Előfizetők.Előfizető.Telefon jelölést, azt kapjuk, hogy* $t_1.p = t_2.p = \perp$, *azaz,* $t_1.S_1 = t_2.S_1 = (Családi, Extra, \perp)$ *de* $25 = t_1.S_2 \neq t_2.S_2 = 65$.

Az XFD-nek megfelelő reguláris FD-t az alábbiak szerint állíthatjuk össze:

$RFD: (\{Dsl, Iptv, Telefon\}, (\{\}, \{\})) \rightarrow (\{Díj\}, (\{\}, \{\}))$

9.3. Példa. *Módosítsuk a fenti DTD-t kismértékben úgy, hogy*

```
<!ELEMENT Előfizető (Név,T_év,(V_év,V_kód)*,Szolgáltatás,Díj)>
<!ELEMENT Szolgáltatás (Dsl,Iptv?,Telefon?)>
```

Legyen t_1 az e_1 csúcsra illeszkedő *tree tuple* (a *Kiss* nevű előfizető részfája), akkor:
t_1(Előfizetők)=e_0
t_1(Előfizetők.Előfizető)=e_1
t_1(Előfizetők.Előfizető.Név)=e_4
t_1(Előfizetők.Előfizető.Név.S)=„Kiss"
t_1(Előfizetők.Előfizető.Díj)=e_{14}
t_1(Előfizetők.Előfizető.Díj.S)=25
t_1(Előfizetők.Előfizető.Dsl)=e_7
t_1(Előfizetők.Előfizető.Dsl.S)=„Családi"
t_1(Előfizetők.Előfizető.Iptv)=e_8
t_1(Előfizetők.Előfizető.Iptv.S)=„Extra"
t_1(Előfizetők.Előfizető.Telefon)=$\perp$

és legyen t_2 az e_2 csúcsra illeszkedő *tree tuple* (a *Nagy* nevű előfizető részfája), akkor:

t_2(Előfizetők)=e_0
t_2(Előfizetők.Előfizető)=e_2
t_2(Előfizetők.Előfizető.Név)=e_5
t_2(Előfizetők.Előfizető.Név.S)=„Nagy"
t_2(Előfizetők.Előfizető.Díj)=e_{15}
t_2(Előfizetők.Előfizető.Díj.S)=65
t_2(Előfizetők.Előfizető.Dsl)=e_9
t_2(Előfizetők.Előfizető.Dsl.S)=„Családi"
t_2(Előfizetők.Előfizető.Iptv)=e_{10}
t_2(Előfizetők.Előfizető.Iptv.S)=„Extra"
t_2(Előfizetők.Előfizető.Telefon)=$\perp$

18. ábra. Két *tree tuple* a 14. ábra DTD és XML fa értékeiből

akkor a tree tuple XFD átalakul:

S_1 *:Előfizetők.Előfizető.Szolgáltatás.Dsl,*
Előfizetők.Előfizető.Szolgáltatás.Iptv,
Előfizetők.Előfizető.Szolgáltatás.Telefon $\rightarrow$ S_2 *: Előfizetők.Előfizető.Díj*

Az alábbi reguláris FD ugyanazt az épségi megszorítást írja le:

$RFD: (\{Szolgáltatás\}, (\{\}, \{\})) \rightarrow (\{Díj\}, (\{\}, \{\}))$

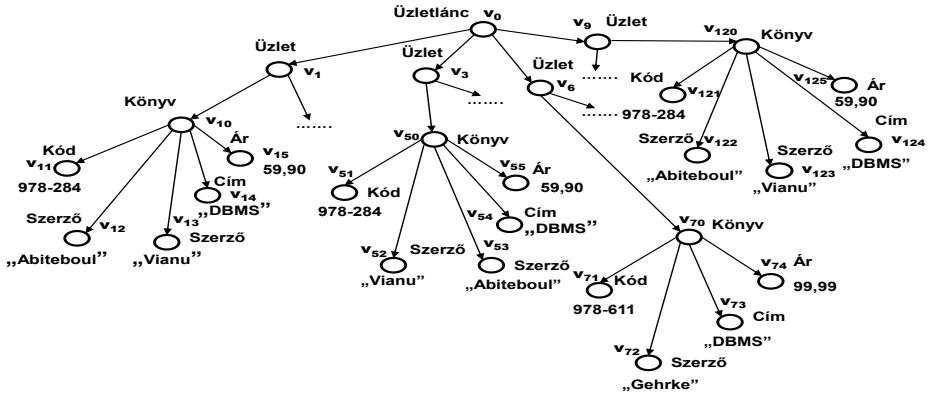
Ez a séma csak akkor felel meg a reguláris funkcionális függőség követelményeinek ha a Szolgáltatás összetett elem, mint nemterminális szimbólum a Dsl,Iptv,Telefon elemek összefűzött értékeit veszi fel terminális helyettesítési értékként. Ezen helyettesítésnél, miután a három elemnév nem szerepelhet nemterminális szimbólumként (a $Szolgáltatás \Rightarrow Dsl\,Iptv^{?}\,Telefon^{?}$ *helyettesítési szabály nem illik reguláris grammatikába), így viszont a terminális értékek típus nélkül maradhatnak (mivel Iptv és Telefon opcionálisak), így például az („Alap"„Alap") összetett érték tartozhat a (Dsl,Iptv) de a (Dsl,Telefon) elempárokhoz is, így két sorban a Szolgáltatás attribútum értékei látszólag egyenlőek lehetnek, de típussal ellátva már különböznének (pl. ha az egyikben csak Iptv, a másikban csak Telefon szerepel). Ezt az esetet csak a kiterjesztett környezetfüggetlen nyelvekre alapozott modell tudja kezelni (11. Fejezet).*

Látható, hogy a *tree tuple* modell szerinti XFD a résztvevő elemek teljes leírásával választja ki a résztvevő komponenseket, a reguláris FD viszont összetett értékű adatokkal dolgozik.

Egy *tree tuple* modell szerinti XFD olyan útkifejezésekből áll, amelyeket egy DTD fa valamennyi lehetséges útkifejezéséből választunk ki. Ha egy ilyen XFD valamennyi útkifejezésének utolsó előtti tagja azonos (ez a feltétel a gyakorlatban előforduló esetekben többnyire teljesül), akkor az XFD által leírt épségi megszorítás RFD-vel is megfogalmazható.

Miután az RFD hatóköre egyetlen reguláris kifejezés (azaz, egy különálló elemtípus deklaráció egy DTD-ben), ha azt akarjuk, hogy a *tree tuple* koncepció kifejező erejét elérjük, a reguláris FD modelljét ki kell bővítenünk, például a reguláris nyelv helyett kiterjesztett környezetfüggetlen nyelvet véve alapul.

Egy másik apró, de lényeges különbség a két modell között szemléltethető a $B\,C^*$ reguláris kifejezés segítségével. Mindkét modellben leírható a $B \rightarrow C$ informális FD, XFD vagy RFD szintaxist használva, de a függőségek



19. ábra. XML példa dokumentum az általánosított *tree tuple* modellhez

értelmezése (szemantikája) különböző lesz. A BC_1C_2 elemekből álló XML dokumentum (fa) megsérti az XFD-t, mert két *tree tuple* sort, $B\,C_1$-et és $B\,C_2$-t tartalmaz, amelyekben a bal oldalak (rendre B) megegyeznek, míg a jobb oldalak (C_1 és C_2), különböznek. Másrészről, az előfordulás kielégíti az RFD-t, mivel csak egyetlen *reguláris* sort, $B\,C_1\,C_2$-t tartalmaz.

9.2. Reguláris FD összehasonlítása az általánosított „tree tuple" XFD-vel

Yu és Jagadish [58] azt találták, hogy a *tree tuple* modell (Arenas és Libkin [6]) nem képes kezelni azokat a megszorításokat amelyekben egyforma típusú testvér elemek egy csoportja együtt, egy komponenst alkotva szerepelnek. A 9.1. Fejezet utolsó bekezdése ugyanerre az esetre világít rá a *tree tuple* XFD és RFD összevetésében: RFD a csoportot egységként, míg XFD elemenként kezeli. Yu és Jagadish kiterjesztették a *tree tuple* modellt a *set* (többszörös, ismétlődő elemek) fogalmával, beleépítve ezeket az XFD specifikációba. Mivel ez a koncepció a *tree tuple* modellt általánosítja, RFD nyilvánvalóan nem alkalmas ezen XML FD-k megvalósítására az általános esetben (azaz, ha az XFD-t alkotó útkifejezések különböző hosszúságúak). A továbbiakban összehasonlítjuk a két koncepciót az általánosított *tree tuple* modell bemutatására használt, publikált példán, amely az RFD-vel is jól kezelhető.

A 19. ábrán mutatott XML dokumentum megfelel az alábbi sémának:

```
Üzletlánc: Rcd
Üzlet: SetOf Rcd
```

```
Könyv: SetOf Rcd
Kód: str
Szerző: SetOf str
Cím: str
Ár: float
```

Ezt a sémát az általánosított *tree tuple* modell leírásához kifejlesztett séma-leíró nyelv szerint kódoltuk. *Rcd* megfelel az XML-séma *complex type*-nak (összetett típus) és *SetOf* azonos az XML-séma $maxOccurs > 1$ kifejezésével, bár, az általánosított *tree tuple* modellben az elemek rendezettségét nem veszik figyelembe.
Yu és Jagadish megfogalmaznak négy informális épségi megszorítás példát [58]. Ezekből három RFD-vel is megvalósítható:

Megszorítás 1: Valahányszor két *Könyv* elem (például a v_{10} és v_{50} csúcsok) megegyező *Kód* értékeket tartalmaz, akkor a *Cím* értékük is azonos.
Megszorítás 2: Valahányszor két azonos nevű *Üzlet*-ben két *Könyv* elem megegyező *Kód* értékeket tartalmaz, akkor az *Ár* értékük is azonos.
Megszorítás 3: Valahányszor két *Könyv* elemben a *Kód* értékek azonosak, akkor a hozzájuk tartozó *Szerző* csoport (*set*) is azonos.
Megszorítás 4: Valahányszor két *Könyv* elemhez azonos *Szerző* csoport és azonos *Cím* érték tartozik, akkor a *Kód* értékek is egyenlőek.

Megszorítás 2 nem írható le a reguláris FD modellel mert különböző hatókörbe (*Üzlet*) tartozó elemekre hivatkozik. Az általánosított *tree tuple* modell a többi Megszorításokat az alábbi XML FD-kkel realizálja:

$FD_1 : \{./Kód \rightarrow ./Cím\}$
$FD_3 : \{./Kód \rightarrow ./Szerző\}$
$FD_4 : \{./Szerző, ./Cím \rightarrow ./Kód\}$

Ezeket az FD-ket a *Könyv* elem hatókörében fogalmazták meg, az XPATH-ból adoptált „." lépés, amely az XPATH *self* fogalmának felel meg, a hatókör szintjére utal. Például, $./Kód$ az FD_1-ben a $/Üzletlánc/Üzlet/Könyv/Kód$ abszolút útkifejezést jelenti.
A 19. ábra XML dokumentuma kielégíti mindhárom XML FD-t, például a v_{10} és v_{50} csúcsokra illeszkedő általánosított *tree tuple* sorok (mondjuk t_1 és t_2) kielégítik FD_3-at, mivel megegyeznek $./Kód$-ban (=978-284) és a megfelelő *Szerző* csoportok ($\{Abiteboul, Vianu\}$ és $\{Vianu, Abiteboul\}$) mint *set*-ek (azaz halmazként, a sorrendre tekintet nélkül) megegyeznek.

A megfogalmazandó RFD-k bázisaként az előbbi sémát reguláris kifejezéssé formáljuk:

$$\left(Kód, Szerző^*, Cím, Ár\right)$$

Az RFD szintaxisában, az FD_1, FD_3, FD_4 XML függőségek alakja:

$$RFD_1 : (\{Kód\}, (\{\}, \{\})) \rightarrow (\{Cím\}, (\{\}, \{\}))$$
$$RFD_3 : (\{Kód\}, (\{\}, \{\})) \rightarrow (\{Cím\}, (\{Szerző\}, \{\}))$$
$$RFD_4 : (\{Cím\}, (\{Szerző\}, \{\})) \rightarrow (\{Kód\}, (\{\}, \{\}))$$

A 19. ábra XML dokumentuma kielégíti RFD_1-et, de megsérti RFD_3-at és RFD_4-et, mivel a $(Abiteboul\, Vianu)$ és $(Vianu\, Abiteboul)$ sorozatok nem egyenlőek: RFD a szimbólumok (elem értékek) rendezett sorozatait kezeli. A 10.3. Fejezetben visszatérünk erre a példára (10.3. Példa), az RFD halmazkonstruktorral kiegészített változata az általánosított *tree tuple* modellel ekvivalens módon értékeli ki a fenti RFD-ket.

9.3. Reguláris FD összehasonlítása a „closest node" XFD modellel

Vincent és munkatársai [53, 54] olyan esetekkel foglalkoztak, amelyek a *tree tuple* modellel nem kezelhetőek, és ezen esetekre megalkották a *closest node* koncepcióját. Funkcionális függőséget definiáltak XML dokumentumokra (a fa szerkezetre építve), és DTD-ket csak abból a célból használtak, hogy kimutassák modelljük ekvivalenciáját DTD jelenléte esetén a *tree tuple* koncepcióval, feltéve, hogy a szóbanforgó XML fa *teljes* (a *teljes XML* definícióját [54]-ben találjuk).
Ebben a modellben a csúcsokon értelmezett *closest()* bool függvény definíciója alapvető fontosságú [54]:

9.1 Definíció
Legyen D egy DTD, legyen T egy D-nek megfelelő XML fa, és legyen v_i és v_j két tetszőleges (nem szükségképpen különböző) cúcs $nodes\,(T)$-ben. A $closest\,(v_i, v_j)$ bool függvény értéke legyen igaz ha létezik egy $x_i^j \in nodes\,(T)$ csúcs úgy, hogy $x_i^j \in aancestor\,(v_i)$ és $x_i^j \in aancestor\,(v_j)$, ahol $aancestor\,(v) = \{v\} \cup ancestor\,(v)$.

A *closest node* teljes definíciója [54]-ban található.

Első ránézésre a *closest node* XFD nem összehasonlítható az RFD-vel, mivel a 9.1. Definíció az *ős* (ancestor) csúcsra utal (egy olyan csúcsra amely

a hivatkozott csúcstól a gyökérig vezető, „felmenő" úton található), azaz, egy a hivatkozott csúcshoz képest „vertikális" irányban lévő elemre és ez a fogalom nem fér össze a reguláris funkcionális függőség „horizontális" jellegével. Azonban, ez a közös *ős* (ancestor) elem lehet az a fő elem (és, a gyakorlatban legtöbbször az is) amelynek a gyermekeiből választjuk az RFD komponenseit. Így a két modell általában jól összehasonlítható.

Kölcsönveszünk egy publikált példát [54], hogy azon demonstráljuk a *closest node* és reguláris FD kifejező erejét.

9.4. Példa. *Ez a példa az alábbi DTD-vel leírt Publikációk adatbázisra épül. Egy, a DTD-nek megfelelő XML fa (adatbázis előfordulás) látható a 20. ábrán.*

```
<!DOCTYPE Gyökér [
<!ELEMENT Gyökér (Publikáció*)>
<!ELEMENT Publikáció (Cím, Szerző*, Kiadó)>
<!ELEMENT Cím  (#PCDATA)>
<!ELEMENT Szerző (#PCDATA)>
<!ELEMENT Kiadó (#PCDATA)>
]>
```

Először megadunk egy a closest node koncepció szerint összeállított XFD-t:

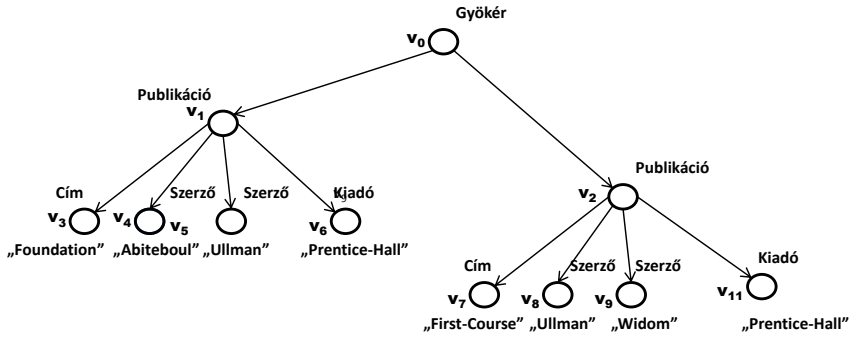
$FD_5 : Gyökér.Publikáció.Kiadó \rightarrow_c Gyökér.Publikáció.Cím$

A 20. ábra dokumentumán closest (v_3, v_6)*=igaz, és closest* (v_7, v_{11})*=igaz, így a closest node modellben két sort kapunk, ezeken ellenőrizhetjük az XFD teljesülését. A bal oldalakra a* $val\,(v_6)=Prentice-Hall=val\,(v_{11})$ *egyenlőséget kapjuk, viszont a jobb oldalak különböznek:* $val\,(v_3) = Foundation \neq First--Course = val\,(v_7)$*, azaz, az előfordulás nem elégíti ki a függőséget.*

A függőség leírható az RFD szintaxisával is:

$RFD_5 : (\{Kiadó\}, (\{\}, \{\})) \rightarrow (\{Cím\}, (\{\}, \{\}))$

A 20. ábra XML formában megjelenített adatainak, mint egy reguláris nyelv mondatainak generálásakor a Publikáció elem deklarációjának megfelelően egy duális mondat is létre jön:



20. ábra. XML dokumentum a *closest node* modell szemléltetésére

$\mathrm{w} = \lfloor Cím \mid Szerző \mid Szerző \mid Kiadó \rceil$
RFD_5 *a* $\mathrm{w}[X] = Kiadó$, $\mathrm{w}[Y] = Cím$ *részsorozatokat választja ki a* w *duális mondatból. Vizsgáljuk meg a 20. ábra XML dokumentumának két adatsorát (a reguláris nyelv mondatait):*

$t_1 = \lfloor Foundation \mid Abiteboul \mid Ullman \mid Prentice-Hall \rceil$
$t_2 = \lfloor First-Course \mid Ullman \mid Widom \mid Prentice-Hall \rceil$

Az előfordulás megsérti az RFD_5 *is:* $t_1[X] = Prentice-Hall = t_2[X]$, *de* $t_1[Y] = Foundation \neq First-Course = t_2[Y]$.

Definiálhatunk egy másik, a reguláris funkcionális függőség szellemének jobban megfelelő RFD-t is:

$$RFD_6 : (\{Kiadó\}, (\{\}, \{\})) \rightarrow (\{\}, (\{Szerző\}, \{\}))$$

Az előfordulás RFD_6*-ot is megsérti:*
$t_1[X] = Prentice-Hall = t_2[X]$, *de* $t_1[Y] = \lfloor Abiteboul \mid Ullman \rceil \neq \lfloor Ullman \mid \mid Widom \rceil = t_2[Y]$.

10. Az RFD korlátozásainak feloldása

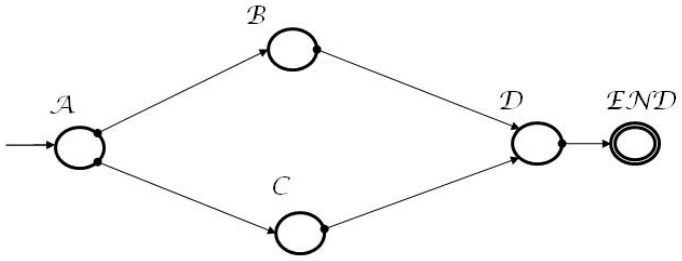
A reguláris funkcionális függőség tárgyalásánál tettünk néhány megkötést, elsősorban azért, hogy az üres jelsorozatok egyenlőségének kérdését az RFD logikai implikációjával kapcsolatban ne kelljen vizsgálni, rögzített sorrendű listákat kezeltünk, nem használtunk számlálókat, mindig feltételeztük, hogy a terminális szimbólumok (a reguláris nyelvet felismerő automata átmenetei) véges sokan vannak. Ebben a fejezetben ezeket a megkötéseket és feloldásuk hatását elemezzük.
Az RFD leírásában öt fontos megkötést tettünk:

- az (1) feltétel (M1)
- kiterjesztett reláció előfordulása egyetlen duális mondaton (M2)
- az attribútumok sorrendje rögzített (M3)
- az RFD (az indexelt eseten kívül) nem használ számlálókat (M4)
- a terminális szimbólumok száma véges (M5)

10.1. RFD az (1) feltétel nélkül

A relációs funkcionális függőségek (FD-k) és a reguláris funkcionális függőségek logikai implikációja végesen axiomatizálható lényegében ugyanazokat az Armstrong típusú axiómákat felhasználva. Az XML funkcionális függőségek (XFD) logikai implikációja általában végesen nem axiomatizálható([6]), a probléma okozója az XFD fa-szerkezete és a korlátlan számú diszjunkció egyidejű jelenléte. Az (1) Feltétel megkövetelésével a diszjunkciót kizártuk. Nézzük meg, érvényesek maradnak-e az RFD-re kimondott állítások ezen megkötés feloldása esetén.

10.1. Példa. *A 21. ábra G gráfja az $A \circ (B + C) \circ D$ reguláris kifejezés által generált nyelvet elfogadó véges automatát reprezentálja. (Ebben a példában a kijelölések leírására egyszerűsített szintaxist használunk: az üres gráfok jelzését elhagyjuk, így (A_1, A_2)=$(\{A\}, (\{\}, \{\}))$ helyett az egyenértékű A kifejezést használjuk.) Legyen $\Sigma = \{A \to B, B \to C, C \to D\}$ RFD-k egy halmaza és legyen $\sigma = \{A \to D\}$ egy RFD G felett. Σ nem teljesíti az (1) feltevést, mivel csak két mondatot tartalmaz a duális nyelv, nevezetesen ABD-t és ACD-t, és egyikük sem foglalja magában $B \to C$-t. A 6.1. Algoritmus igaz eredményt ad a (Σ, σ, G) inputtal: induláskor az A-k egyszínűek („zöldek") a két gráfon,*



21. ábra. Diszjunkciós reguláris kifejezés véges automatája

a B, C, D csúcsok a két gráfon eltérő színűek. $A \to B$ zöldre színezi B-t, majd $B \to C$ a C-t, végül $C \to D$ az $A \to D$ kielégítéséhez szükséges D-t is. Ha ϵ-t sorokból való vetítésnél érvényes értéknek fogadjuk el, akkor könnyű megmutatni, hogy $\Sigma \models \sigma$, azaz, a 6.1. Algoritmus helyesen működik. Ha ugyanis egy I előfordulásra $I \models \Sigma$, akkor három eset lehetséges:

1. $schema\,(I) = ABD$
2. $schema\,(I) = ACD$
3. $schema\,(I) = ABD, ACD$ *(az M2 nélkül, 10.1. Megjegyzés)*

Az 1. esetben $t\,[C] = \epsilon; t \in I$, azaz bármely $t_1, t_2 \in I$-re $t_1\,[C] = t_2\,[C] = \epsilon$, akkor mivel $I \models C \to D$, következik, hogy $t_1\,[D] = t_2\,[D]$, tehát $I \models A \to D$.
A 2. esetben $t\,[B] = \epsilon; t \in I$, azaz bármely $t_1, t_2 \in I$-re $t_1\,[C] = t_2\,[C]$ mivel $I \models B \to C$, azaz, $I \models C \to D$-ből következik, hogy $t_1\,[D] = t_2\,[D]$, tehát $I \models A \to D$.
A 3. esetben (ha az M2 Megkötéstől is eltekintünk) bármely olyan $t_1, t_2 \in I$-re, amelyekre $type\,(t_1) = type\,(t_2)$ az előzőek szerint teljesül, hogy $t_1\,[D] = t_2\,[D]$. Ha viszont $type\,(t_1) \neq type\,(t_2)$, akkor legyen (mondjuk) $type\,(t_1) = ABD$, akkor $I \models A \to B$ és $t_1\,[B] \neq t_2\,[B] = \epsilon$ miatt $t_1\,[A] \neq t_2\,[A]$, tehát $I \models A \to D$.

Úgy látszik, hogy a logikai implikáció a diszjunktív függőségekre is kvadratikus időben eldönthető, annak ellenére, hogy az általános esetben ez coNP teljes [6]. Az RFD általános tárgyalásánál azonban kikötöttük az (1) feltétel teljesülését, azért, hogy elkerüljük az üres jelsorozatok összehasonlítását.

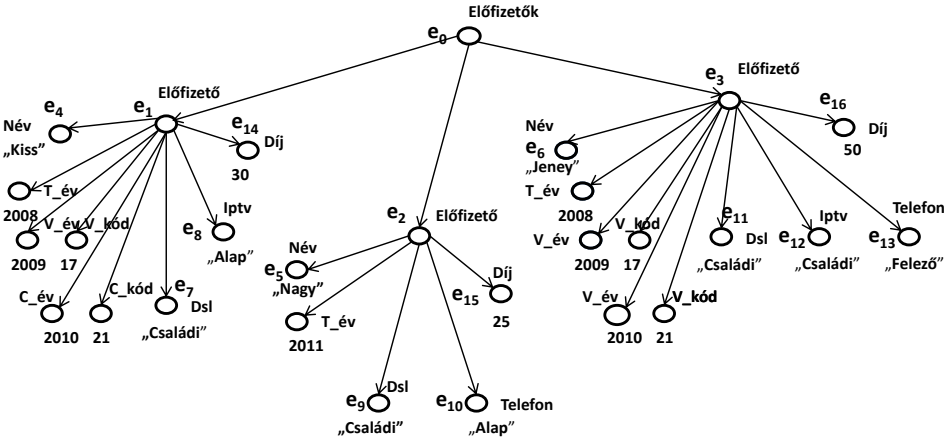
10.1 Megjegyzés (RFD implikációjának eldöntése M1 nélkül)
Az M1 Megkötés elhagyása esetén az RFD logikai implikációját eldöntő 6.1. Algoritmus nyilván helyes döntést hoz ha az implikáció nem teljesül: az algoritmus végén az ellenpélda egy olyan előfordulást alkot, amelynek két sora a függőséghalmaz valamennyi függőségét kielégíti, de az implikálandó függőséget nem. Ha azonban az algoritmus eredménye az implikáció teljesülését állítja, akkor ez az eredmény csak úgy vihető át az M1 Megkötés feloldásával értelmezett előfordulásokra, ha az üres jelsorozatokat is érvényesnek fogadjuk el, azaz, ha $I \models \Sigma$ *és* $U \to W \in \Sigma$*, valamint* $t_1, t_2 \in I$ *és* $t_1[U] = t_2[U] = \epsilon$*, akkor* $t_1[W] = t_2[W] = \omega$ *(lehet, de nem szükségszerű, hogy* $\omega = \epsilon$*).*
A 10.1. Példa első két eseténél feltételeztük az M2 megkötés érvényesülését, ettől azonban el is tekinthettünk: bármely $t_1, t_2 \in I$*-re, ha* $type(t_1) \neq type(t_2)$ *(mondjuk* $type(t_1) = ABD$*,*$type(t_2) = ACD$*), akkor* $I \models A \to B$ *miatt* $t_1[A] \neq$ $= t_2[A]$*, mert* $\epsilon \neq t_1[B] \neq t_2[B] = \epsilon$*, és ezért* t_1, t_2 *irreleváns a* $B \to C$ *függőség szempontjából is, a* $C \to D$ *eset az* $A \to B$ *szimmetrikus párja, azaz* I *a* Σ *szempontjából két diszjunkt, típusában homogén részhalmazra esik szét.*

10.1. Tétel. *Az {RFD-1,RFD-2,RFD-3} axiómarendszer a reguláris funkcionális függőségek implikációjára nézve helyes és teljes akkor is, ha a reguláris kifejezés diszjunkciókat is tartalmaz.*

Bizonyítás
Ezen tétel kontextusában az (1) feltétel érvényességét nem vesszük adottnak. A 10.1. Példa gondolatmenetét követve és a 10.1. Megjegyzést figyelembe véve látjuk, hogy a 6.1. Algoritmus akkor is helyes döntést hoz ha a reguláris kifejezés tetszőleges számú diszjunkciót tartalmaz (a diszjunkciók száma az algoritmusban nem játszik szerepet). Természetesen az M1 megkötést el kell hagynunk (a diszjunkció miatt csak a triviális üres előfordulás lenne megengedett), viszont meg kell engednünk az üres jelsorozatokat, azzal, hogy két üres jelsorozat mindig egyenlő.

10.2. RFD kiterjesztett sémán

Az M2 megkötést a 2.1 és 4.2. Megjegyzések fogalmazták meg. Feloldása esetén a kiterjesztett reláció előfordulása (kiterjesztett előfordulás) a teljes reguláris nyelv egy tetszőleges (véges) részhalmaza, azaz, az előfordulás akár minden mondata különböző típusú lehet, az előfordulás sorai (a reguláris



22. ábra. XML példa dokumentum kiterjesztett sémára

nyelv mondatai) különböző duális mondatokhoz tartozhatnak. Az előfordulás soraihoz tartozó duális mondatok együttesen alkotják az előfordulás sémáját. Ezt a sémát az M2 megkötés alkalmazásától megkülönböztetés céljából kiterjesztett sémának nevezzük.

10.1 Definíció (Kiterjesztett előfordulás)
Legyen L reguláris nyelv amelyet az $M(L)$ automata fogad el, legyen $D(L)$ a társított duális nyelv. $R(L) = \{t | t \in L, type(t) \in D(L)\}$ kiterjesztett reláció L felett. Az R kiterjesztett reláció sémája a társított duális nyelv, azaz $schema(R) = D(L)$. Az $\hat{I}(R) \subseteq_{fin} L$ (vagy egyszerűen $\hat{I}$ ha R egyértelmű a szövegkörnyezetben) véges sorhalmaz $R(L)$-nek egy kiterjesztett előfordulása. A továbbiakban jelöljük $\hat{schema}\left(\hat{I}\right) = \left\{schema(t) | t \in \hat{I}\right\}$-t, és $\hat{\mathcal{I}}(R) = \left\{\hat{I} | \hat{I} \subseteq_{fin} L\right\}$-vel az előfordulások halmazát L felett. $\hat{\mathcal{I}}(L)$ akkor és csak akkor véges ha az L nyelv nem-rekurzív. Ha $t \in \hat{I} \in \hat{\mathcal{I}}(L)$, akkor $type(t) \in \hat{schema}\left(\hat{I}\right) \subseteq_{fin} schema(R)$.

10.2. Példa. *A 7.4. Példában egy nyilvános telefonhálózat előfizetőinek adatain mutattuk be az RFD működését. A kiterjesztett előfordulás szemléltetéséhez kissé kibővítjük az idézett példában használt XML sémát, két új elem (<C_év>, <C_kód>) felvételével (23. ábra). A 22. ábrán ezen XSD egy előfordulását, egy, az XSD-nek megfelelő XML dokumentumot láthatunk, az*

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
               elementFormDefault="unqualified">
  <xs:element name="Előfizetők">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded" minOccurs="1">
        <xs:element name="Előfizető" type="t_Előfizető"
                 maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="t_Előfizető">
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:element name="Név" type="xs:NCName"/>
      <xs:element name="T_év" type="xs:integer"/>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="V_év" type="xs:integer"/>
      <xs:element name="V_kód" type="xs:integer"/>
  </xs:sequence>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="C_év" type="xs:integer"/>
      <xs:element name="C_kód" type="xs:integer"/>
  </xs:sequence>
      <xs:element name="Dsl" type="xs:NCName"/>
      <xs:element minOccurs="0" name="Iptv" type="xs:NCName"/>
      <xs:element minOccurs="0" name="Telefon" type="xs:NCName"/>
      <xs:element name="Díj" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

23. ábra. A kiterjesztett sémájú példa (22. ábra) XSD sémaleírása

XML dokumentum adatsorait mint mondatokat tartalmazó reguláris nyelvet elfogadó véges automatát a 24. ábra mutatja. Egy adott előfizető számára nyújtott szolgáltatások nem szükségképpen teljeskörűek, a Telefon vagy Iptv szolgáltatás igénybe vétele opcionális: lehetséges Dsl-t használni akár (vonalas) Telefon kapcsolat nélkül is, használhatunk Iptv-t vagy eltekinthetünk attól. A Díj elem mutatja az igénybe vett szolgáltatások havonta fizetendő alapdíját. Ez a díj egyedül a nyújtott szolgáltatásoktól függ a 3. táblázat szerint (például, Telefon: Alap=5,Favorit=10,Felező=15 stb.), azaz, ha két előfizető azonos szolgáltatásokat rendelt meg akkor a havidíjuk azonos lesz. A 7.4. Példából felidézve, egy érvényes adatbázis ki kell, hogy elégítse az igénybe vett szolgáltatások és a havonta fizetendő díj között fennálló reguláris funkcionális függőséget:
$RFD_{1b} : (\{Dsl, Iptv, Telefon\}, (\{\}, \{\})) \rightarrow (\{Díj\}, (\{\}, \{\}))$
RFD_{1b} a 22. ábra mindhárom duális mondatából különböző részsorozat-párt választ ki:

$\mathrm{w}_1 = \lfloor Név \mid T\_év \mid V\_év \mid V\_kód \mid C\_év \mid C\_kód \mid Dsl \mid Iptv \mid Díj \rceil$
$t_1 = \lfloor Kiss \mid 2008 \mid 2009 \mid 17 \mid 2010 \mid 21 \mid Családi \mid Alap \mid 30 \rceil$
$\mathrm{w}_1[X] = \lfloor Dsl \mid Iptv \rceil, \mathrm{w}_1[Y] = Díj$
$t_1[X] = \lfloor Családi \mid Alap \rceil, t_1[Y] = 30$
$\mathrm{w}_2 = \lfloor Név \mid T\_év \mid Dsl \mid Telefon \mid Díj \rceil$
$t_2 = \lfloor Nagy \mid 2011 \mid Családi \mid Alap \mid 25 \rceil$
$\mathrm{w}_2[X] = \lfloor Dsl \mid Telefon \rceil, \mathrm{w}_2[Y] = Díj$
$t_2[X] = \lfloor Családi \mid Alap \rceil, t_2[Y] = 25$
$\mathrm{w}_3 =$
$= \lfloor Név \mid T\_év \mid V\_év \mid V\_kód \mid V\_év \mid V\_kód \mid Dsl \mid Iptv \mid Telefon \mid Díj \rceil$
$t_3 = \lfloor Jeney \mid 2008 \mid 2009 \mid 17 \mid 2010 \mid 21 \mid Családi \mid Családi \mid Felező \mid 50 \rceil$
$\mathrm{w}_3[X] = \lfloor Dsl \mid Iptv \mid Telefon \rceil, \mathrm{w}_3[Y] = Díj$
$t_3[X] = \lfloor Családi \mid Családi \mid Felező \rceil, t_3[Y] = 50$

Mivel a duális mondatok különbözőek, a 22. ábra az M2 megkötéssel a kapcsolódó kiterjesztett reláció három előfordulását tartalmazza, ezek mindegyike triviálisan kielégíti RFD_{1b}-t (a 7.4. Példában az RFD_{1a} függőségét úgy tettük érdemivé, hogy az erős és gyenge kiértékelést használtuk). Az M2 megkötés feloldása tágabban értelmezi az előfordulás fogalmát (reguláris mondatok véges halmaza), így a 22. ábra a kiterjesztett reláció egy, három sort tartalmazó előfordulása. Az RFD_{1b} ezen az előforduláson sem az erős, sem a gyenge kiértékeléssel nem fejez ki érdemi megszorítást, mert mind a duális mondatok, mind az azokból az RFD_{1b} bal oldala által kimetszett jelsorozatok páronként különbözőek. Minden megkötés nélkül RFD_{1b}-t a kiterjesztett előfordulás megsérti: $t_1[X] = \lfloor Családi \mid Alap \rceil = t_2[X]$, de $t_1[Y] = 30 \neq 25 = t_2[Y]$. Másrészt,

a 22. ábrán a 3. táblázat szerint a Díj értékek helyesek, így az előfordulás ki kellene, hogy elégítse RFD_{1b}-t. Az ellentmondás nyilvánvalóan abból adódik, hogy a reguláris nyelv szimbólumait típus nélkül hasonlítottuk össze, ezért legalább a gyenge kiértékelésre szükség van az M2 megkötés feloldásakor is.

A 7.3. Példa RFD_{1a} függőségét a 22. ábra dokumentumához igazítva kapjuk

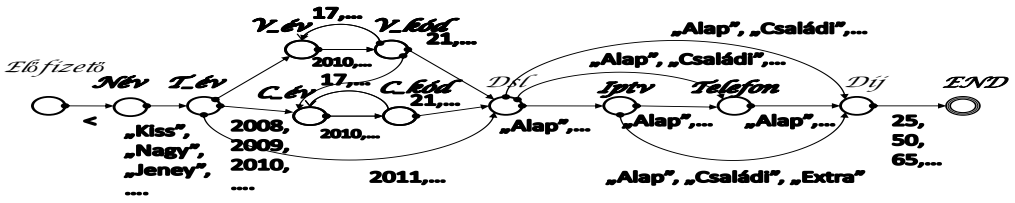
$$RFD_{1c} : (\{T\_év\}, (\{\}, \{\})) \rightarrow$$
$$(\{\}, (\{V\_év, V\_kód, C\_év, C\_kód\}, \{(V\_év, V\_kód, C\_év, C\_kód)\}))$$

Az XML dokumentumban realizálódó adatbázis előfordulás az M2 megkötés feloldás esetén is kielégíti RFD_{1c}-t, mert két olyan sor van, a Kiss és Jeney előfizetők adatai, ahol a függőség bal oldala azonos értékeket tartalmaz (T_év=2008), és itt a jobb oldalak (2009 17 2010 21) is azonosak, ha a típusokat nem vesszük figyelembe. Ebben a példában a gyenge vagy erős kiértékelést használva is ugyanerre az eredményre jutunk, hiszen nemcsak a duális mondatok, hanem a duális mondatokból kimetszett bal oldalak is különbözőek:

$$\mathrm{w}_1[X] = T\_év, \mathrm{w}_1[Y] = \lfloor V\_év \mid V\_kód \mid C\_év \mid C\_kód \rceil$$
$$t_1[X] = 2008, t_1[Y] = \lfloor 2009 \mid 17 \mid 2010 \mid 21 \rceil$$
$$\mathrm{w}_2[X] = T\_év, \mathrm{w}_2[Y] = \epsilon$$
$$t_2[X] = 2011, t_2[Y] = \epsilon$$
$$\mathrm{w}_3[X] = T\_év, \mathrm{w}_3[Y] = \lfloor V\_év \mid V\_kód \mid V\_év \mid V\_kód \rceil$$
$$t_3[X] = 2008, t_3[Y] = \lfloor 2009 \mid 17 \mid 2010 \mid 21 \rceil$$

A példából jól látszik, hogy bár az erős és gyenge kiértékeléssel is kielégíti az előfordulás a függőséget, a típusvizsgálatra szükség van.

10.2 Megjegyzés (RFD implikációjának eldöntése M2 nélkül) *Az M2 Megkötés elhagyása esetén az RFD logikai implikációját eldöntő 6.1. Algoritmus nyilván helyes döntést hoz ha az implikáció nem teljesül: az algoritmus végén az ellenpélda egy olyan előfordulást alkot, amelynek két sora a függőséghalmaz valamennyi függőségét kielégíti, de az implikálandó függőséget nem. Ha azonban az algoritmus eredménye az implikáció fennállását adja, akkor ez az eredmény csak úgy vihető át az M2 Megkötés feloldásával értelmezett előfordulásokra, ha az üres jelsorozatokat is érvényesnek fogadjuk el, azaz, ha $I \models \Sigma$ és $U \rightarrow W \in \Sigma$, valamint $t_1, t_2 \in I$ és $t_1[U] = t_2[U] = \epsilon$, akkor $t_1[W] = t_2[W] = \omega$ (lehet, de nem szükségszerű, hogy $\omega = \epsilon$).*



24. ábra. Kiterjesztett előfordulás példát leíró XML elemtípus FSA gráfja

10.3. RFD lista és halmazkonstruktorokkal

Az $M(L)$ gráf csúcspontjait lista konstruktorokként értelmezve, az RFD modellből egy vele ekvivalens, egyszerűsített összetett értékű adatbázis modellhez jutunk. Ha ezt a modellt halmazkonstruktorokkal kibővítjük teljes értékű összetett értékekkel dolgozó adatbázist kapunk. Az RFD átvihető erre az adatbázis modellre, ha a vetítés és összehasonlítás fogalmait kiterjesztjük. Ez a modell azonban az eredeti RFD koncepciótól túlságosan eltávolodik, ezért itt egyszerűbb megoldást választottunk. Az RFD-t egyrészt listakonstruktorral, másrészt halmazkonstruktorral bővítjük.
Ha az $M(L)$ gráf csúcspontjait lista konstruktorokként értelmezzük, egy adott csúcshoz tartozó listát az $M(L)$ gráf bejárásakor az adott csúcs behelyettesítésével kapott terminális szimbólumok sorozata alkotja. Ha a csúcs nem tartozik körhöz, a lista egy elemű lesz. Ha a csúcs körhöz tartozik, akkor a lista a bejárástól függő hosszúságú. A listát a reguláris nyelv megfelelő mondatából projekcióval kapjuk meg. A funkcionális függőség kielégítettségének eldöntéséhez szükségünk van a listaértékek (a reguláris mondatokból kivetített sorozatok) összehasonlítására.

10.2 Definíció (Reguláris lista konstruktor)
Legyen L reguláris nyelv, v az $M(L)$ gráf egy csúcsa. Legyen $t \in L$ reguláris mondat és legyen type$(t) = \mathrm{w}$. Legyen $\mathrm{w} = v_1\, v_2\, \ldots\, v_k; k \geq 0$ és jelölje $\mathrm{w}[v] = v_{i_1}\, v_{i_2}\, \ldots\, v_{i_v}; 0 \leq i_1 < i_2 < \ldots < i_v \leq k; v_{i_j} = v, i_1 \leq i_j \leq i_v$ a v csúcs előfordulásait a w *duális mondatban, pozíció szerint indexelve. Legyen $t = t_1\, t_2\, \ldots\, t_k$ akkor a v lista konstruktor a t mondatból készített listáját a $t_{list}[v] = t_{i_1}\, t_{i_2}\, \ldots\, t_{i_v}$ terminális jelsorozat adja meg.*

10.3 Definíció (Projekciók szigorú összehasonlítása listaként)
Legyen L reguláris nyelv, v az $M(L)$ gráf egy csúcsa. Legyenek $t^1, t^2 \in L$

mondatok úgy, hogy type $(t^1)=\mathrm{w}^1$ *és type* $(t^2)=\mathrm{w}^2$. *Ha* $\mathrm{w}^1[v]\neq\mathrm{w}^2[v]$, *akkor a két mondaton a v csúcs által konstruált listák is különböznek (jelölve* $t^1_{list}[v]\neq$ $=t^2_{list}[v]$*). Legyen* $\mathrm{w}^1[v]=\mathrm{w}^2[v]=v_{i_1}v_{i_2}\ldots v_{i_v}$ *és legyen* $t^k_{list}[v]=t^k_{i_1}t^k_{i_2}\ldots t^k_{i_v}; k=$ $=1,2$. $t^1_{list}[v]=t^2_{list}[v]$ *akkor és csak akkor, ha* $t^1_{i_j}=t^2_{i_j}; i_1\leq i_j\leq i_v$.

A lista konstruktor segítségével definiálhatjuk az RFD megfelelő kiértékelését is. Legyen L reguláris nyelv, Y kijelölés az $M(L)$ gráf felett, legyenek $t^1,t^2\in L$ mondatok úgy, hogy $type(t^1)=\mathrm{w}^1$ és $type(t^2)=\mathrm{w}^2$. Ha $\mathrm{w}^1[Y]=\mathrm{w}^2[Y]$, legyen $t^1_{list}[Y]=t^2_{list}[Y]$ akkor és csak akkor, ha $\forall v\in Y$ csúcsra $t^1_{list}[v]=t^2_{list}[v]$ teljesül. Azt mondjuk, hogy az I előfordulás lista konstruktor szerinti kiértékeléssel kielégíti az $X\rightarrow Y$ RFD-t, ha bármely két $t^1,t^2\in I$ mondatra $t^1_{list}[X]=t^2_{list}[X]$ csak akkor állhat fenn, ha $t^1_{list}[Y]=t^2_{list}[Y]$ is igaz. Könnyen belátható, hogy ez a kiértékelési meghatározás az eredeti RFD definícióval az előfordulások halmazára nézve ekvivalens.

10.4 Definíció (Reguláris halmaz konstruktor)
Legyen L reguláris nyelv, v az $M(L)$ *gráf egy csúcsa. Legyen* $t\in L$ *reguláris mondat és legyen type* $(t)=\mathrm{w}$. *Legyen* $\mathrm{w}=v_1\,v_2\,\ldots\,v_k; k\geq 0$ *és jelölje* $\mathrm{w}[v]=$ $=v_{i_1}\,v_{i_2}\,\ldots\,v_{i_v}; 0\leq i_1<i_2<\ldots<i_v\leq k; v_{i_j}=v, i_1\leq i_j\leq i_v$ *a v csúcs előfordulásait a* w *duális mondatban, pozíció szerint indexelve. Legyen* $t=$ $=t_1\,t_2\,\ldots\,t_k$ *akkor a v halmaz konstruktor a t mondatból készített halmazát a terminális szimbólumok* $t_{set}[v]=\{t_{i_1},t_{i_2},\ldots,t_{i_v}\}$ *halmaza adja meg.*

10.5 Definíció (Projekciók szigorú összehasonlítása halmazként)
Legyen L reguláris nyelv, Y egy kijelölés $M(L)$ *felett. Legyenek* $t^1,t^2\in L$ *úgy, hogy type* $(t^1)=\mathrm{w}^1$ *és type* $(t^2)=\mathrm{w}^2$. *Legyen* $\mathrm{w}^1[Y]=\mathrm{w}^2[Y]=v_1\,v_2\,\ldots\,v_k; k\geq$ ≥ 0 *és legyen nodes* $(\mathrm{w}^1[Y])=nodes(\mathrm{w}^2[Y])=\{v|\exists j\in[0,k], v=v_j\}$. *Legyen* $v\in nodes(\mathrm{w}^i[Y])$ *és legyen* $t^i_{set}[v]=\{t^i[v_j]\,|\exists j\in[0,k], v=v_j\}; i=1,2$. *Azt mondjuk, hogy* $t^1_{set}[Y]$ *halmazként szigorúan egyenlő* $t^2_{set}[Y]$*-nal (jelölve* $t^1_{set}[Y]\simeq t^2_{set}[Y]$*), ha* $t^1_{set}[v]=t^2_{set}[v]; \forall v\in nodes(\mathrm{w}^1[Y])$ *(az azonos szimbólumokat gyakoriságuk szerint véve figyelembe). Ha* $\mathrm{w}^1[Y]=\mathrm{w}^2[Y]=\epsilon$, *akkor* $t^1_{set}[Y]\simeq t^2_{set}\{Y\}$.

10.3. Példa. *A 9.2. Fejezetben egy példán hasonlítottuk össze az RFD modellt az általánosított tree tuple modellel. A példában felhasznált XML dokumentum a 19. ábrán látható, a 25. ábra az XML dokumentum XML séma leírását mutatja. Az XML dokumentum adatait a szokásos reguláris/duális nyelv formátumban összefoglalva:*

$\mathrm{w}_1=\lfloor Kód\,|\,Szerző\,|\,Szerző\,|\,Cím\,|\,Ár\rceil$
$t_1=\lfloor 978-284\,|\,Abiteboul\,|\,Vianu\,|\,DBMS\,|\,59{,}90\rceil$

$\mathrm{w}_2 = \lfloor Kód \mid Szerző \mid Szerző \mid Cím \mid Ár \rceil$
$t_2 = \lfloor 978-284 \mid Vianu \mid Abiteboul \mid DBMS \mid 59{,}90 \rceil$
$\mathrm{w}_3 = \lfloor Kód \mid Szerző \mid Cím \mid Ár \rceil$
$t_3 = \lfloor 978-611 \mid Gehrke \mid DBMS \mid 99{,}99 \rceil$
$\mathrm{w}_4 = \lfloor Kód \mid Szerző \mid Szerző \mid Cím \mid Ár \rceil$
$t_4 = \lfloor 978-284 \mid Abiteboul \mid Vianu \mid DBMS \mid 59{,}90 \rceil$

Felidézzük a 9.2. Fejezetből az általánosított tree tuple modellben jól leírható épségi megszorírásokat:

Megszorítás 1: Valahányszor két Könyv elem (például a v_{10} és v_{50} csúcsok) megegyező Kód értékeket tartalmaz, akkor a Cím értékük is azonos.
Megszorítás 3: Valahányszor két Könyv elemben a Kód értékek azonosak, akkor a hozzájuk tartozó Szerző csoport (set) is azonos.
Megszorítás 4: Valahányszor két Könyvs elemhez azonos Szerző csoport és azonos Cím érték tartozik, akkor a Kód értékek is egyenlőek.

Az RFD szintaxisában ezeket a megszorításokat a következőképpen fogalmaztuk meg:

$RFD_1 : (\{Kód\}, (\{\}, \{\})) \rightarrow (\{Cím\}, (\{\}, \{\}))$ *($X_1 \rightarrow Y_1$)*
$RFD_3 : (\{Kód\}, (\{\}, \{\})) \rightarrow (\{Cím\}, (\{Szerző\}, \{\}))$ *($X_3 \rightarrow Y_3$)*
$RFD_4 : (\{Cím\}, (\{Szerző\}, \{\})) \rightarrow (\{Kód\}, (\{\}, \{\}))$ *($X_4 \rightarrow Y_4$)*

A 19. ábra XML dokumentuma kielégíti RFD_1-et, de megsérti RFD_3-at és RFD_4-et, mivel a $(Abiteboul\, Vianu)$ és $(Vianu\, Abiteboul)$ sorozatok nem egyenlőek: RFD a szimbólumok (elem értékek) rendezett sorozatait kezeli. Az RFD halmazkonstruktorral kiegészített változata az általánosított tree tuple modellel ekvivalens módon értékeli ki a fenti RFD-ket:

$\mathrm{w}_1[X_3] = \mathrm{w}_3[X_3] = \mathrm{w}_4[X_3] = Kód, \mathrm{w}_1[Y_3] = \lfloor Cím \mid Szerző \mid Szerző \rceil$
$\mathrm{w}_2[X_3] = Kód, \mathrm{w}_2[Y_3] = \lfloor Cím \mid Szerző \mid Szerző \rceil$

Látható, hogy az előfordulás kielégíti RFD_3-at, mert a három 978-284 Kód-értékű könyv $t_{set}[Y_3]$ halmazai azonosak: $\{\{DBMS\}, \{Abiteboul, Vianu\}\}$.

$\mathrm{w}_1[X_4] = \mathrm{w}_3[X_4] = \mathrm{w}_4[X_4] = \lfloor Cím, \mid Szerző \mid Szerző \rceil$
$\mathrm{w}_1[Y_4] = Kód$
$\mathrm{w}_2[X_4] = \lfloor Cím \mid Szerző \rceil, \mathrm{w}_2[Y_4] = Kód$

Az előfordulás RFD_4-et is kielégíti, mert három könyv Cím,Szerző halmazértéke, $t_{set}[X_4]$ azonos, mindegyik $\{\{DBMS\}, \{Abiteboul, Vianu\}\}$, és ezeken a sorokon a jobboldal, Kód mező halmazértékei, $t_{set}[Y_4]$ is azonosak, mindegyik $\{978-284\}$.

10.6 Definíció (Projekciók gyenge összehasonlítása halmazként) *Legyen L reguláris nyelv, Y egy kijelölés $M(L)$ felett. Legyenek $t^1, t^2 \in L$ úgy, hogy $type(t^1) = type(t^2) = \mathrm{w}$. Legyen $\mathrm{w}[Y] = v_1 v_2 \ldots v_k; k \geq 0$ és $t^i_{set}\{Y\} = \{t^i[v_j] \,|\exists j \in [0,k], v = v_j\}; i = 1,2$ és legyen r^i a $t^i_{set}\{Y\}$ terminális szimbólumhalmaz különböző szimbólumainak a száma. Azt mondjuk, hogy $t^1_{set}\{Y\}$ halmazként gyengén egyenlő $t^2_{set}\{Y\}$-nal (jelölve $t^1_{set}\{Y\} \approx t^2_{set}\{Y\}$), ha $r^1 = r^2$, és $t^1_{set}[v] = t^2_{set}[v]; \forall v \in nodes(\mathrm{w}^1[Y])$ (az azonos szimbólumok megkülönböztetése nélkül). Ha $\mathrm{w}[Y] = \epsilon$, akkor $t^1_{set}\{Y\} \approx t^2_{set}\{Y\}$.*

Mivel mindkét halmazkonstruktoros modell a nemterminális szimbólumokhoz (az $M(L)$ gráf csúcsihoz) rendel összetett értékeket és ezek összehasonlítását definiálja, a 6.1. Algoritmus az RFD-k logikai implikációját halmazkonstruktoros kiértékelés esetén is helyesen dönti el. A 10.3. Példából következik, hogy egy adott előfordulás az RFD eredeti definíciójában szereplő kiértékelés alkalmazásával kielégíthet egy RFD-t míg a halmazkonstruktoros kiértékeléssel nem, és ugyanez fordítva is igaz.

10.4. RFD bővítése numerikus megszorításokkal

Az RFD modell jól kezeli az XML sémanyelv DTD reguláris kifejezéseit, de a W3C XML-séma (*XSD Mixed Type*, *XSD Element Substitution*, *XSD Restriction* etc.) megszorításokat nem tudja értelmezni. De az RFD modell az XML-séma numerikus megszorításaival bővíthető, hasonlóan a [30] cikkben alkalmazott módszerhez. Az RFD definíciójában az $M(L)$-on tett kijelölések között adtuk meg a funkcionális függőségi kapcsolatot: a kijelölés a gráf csúcsaiból (nem ismétlődő rész) és a tranzitív lezárt egy részgráfjából (ismételhető rész) állott. A gráf minden elfogadó bejárásából (duális mondat) kiválasztott a kijelölés egy részsorozatot, ez a részsorozat a kijelölés minden szimbólumához egy számlálót rendel. A számláló 0 vagy 1 a nem ismétlődő részen és $n \geq 0$ az ismételhető részen (n az ismétlések száma). Ezekkel a számlálókkal értelmezhetjük a reguláris funkcionális függőség egy változatát.

Egy kijelölés egy adott duális mondatból egyértelműen kiválaszt egy részsorozatot. Megadtunk két különböző módszert amelyekkel a kijelölés (a módszerre nézve egyértelműen) választ ki egy részsorozatot. A kiválasztás eredményeképpen kapott részsorozathoz számlálók sorozatát rendeljük.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                      elementFormDefault="qualified">
  <xs:element name="Üzletlánc">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" ref="Üzlet"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Üzlet">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" ref="Könyv"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Könyv">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Kód"/>
        <xs:element maxOccurs="unbounded" ref="Szerző"/>
        <xs:element ref="Cím"/>
        <xs:element ref="Ár"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Kód" type="xs:NMTOKEN"/>
  <xs:element name="Szerző" type="xs:NCName"/>
  <xs:element name="Cím" type="xs:NCName"/>
  <xs:element name="Ár" type="xs:string"/>
</xs:schema>
```

25. ábra. Az általánosított *tree tuple* modell példa (19. ábra) XSD sémaleírása

10.7 Definíció (Részsorozat számlálói)
Legyen Γ *véges ábécé és legyen* $\mathrm{w} \in \Gamma^*$ *úgy, hogy* $\mathrm{w} = v_1 v_2 \ldots v_n; n \geq 0$. *Legyen* $v \in [\mathrm{w}]$, *akkor* $\exists 1 \leq i_1 < i_2 < \ldots < i_k \leq n$ *úgy, hogy* $v_i = v$ *ha* $i \in \{i_1, i_2, \ldots, i_k\}$ *és* $v_i \neq v$ *egyébként. A* v *szimbólum* w*-re vonatkoztatott számlálója a* k *szám, jelölve* $count\,(\mathrm{w}/v) = k$. *Ha* $v \notin [\mathrm{w}]$ *akkor* $count\,(\mathrm{w}/v) = 0$.

A 10.7. Definícióból következik, hogy ha $\mathrm{w} \in \Gamma^*$, akkor $\sum_{v \in [\mathrm{w}]} count\,(\mathrm{w}/v) = |\mathrm{w}|$, valamint, hogy ha $\mathrm{w} = \epsilon$, akkor $\forall v \in \Gamma, count\,(\mathrm{w}/v) = 0$.

Legyen $\mathrm{w} = v_1\, v_2\, \ldots\, v_n$ duális mondat (w szimbólumsorozat) az $M\,(L) = (V, E)$ gráf felett.
Jelölje $walk\,(\mathrm{w}) = (\mathcal{START}, v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n, \mathcal{END})$
(rövidítve $walk\,(\mathrm{w}) = (v_1, v_2, \ldots, v_n)$) $M\,(L)$-nek a w-t előállító bejárását.

10.8 Definíció (Kiválasztott részsorozat számlálója)
Legyen $Y = (Y_1, Y_2)$ *egy kijelölés és* w *duális mondat* $M\,(L)$ *felett. Legyen* $walk\,(\mathrm{w}) = (v_1, v_2, \ldots, v_n)$. Y_1 *szimbólumai bejárásuk szerint kerültek kiválasztásra (ha a bejárás érintette őket). Minden* $e \in E\,(Y_2)$ *él esetén, az él végpontjai az elérés sorrendjében kerültek kiválasztásra (ha a bejárás érintette őket). A kiválasztási folyamat végére a* w *duális mondatból kiválasztott szimbólumokból felépült az (esetleg üres)* $\mathrm{w}\,[Y] = v_{i_1} v_{i_2} \ldots v_{i_k}$ *(*$1 \leq i_1 < i_2 < \ldots < i_k \leq n\,(k \geq 0)$*) sorozat. Az* Y *kijelölés* w*-re vonatkoztatott számlálója a* $\mathrm{w}\,[Y]$ *különböző szimbólumaiból és azok* $\mathrm{w}\,[Y]$*-ra vonatkoztatott számlálóiból képzett párok halmaza (*$count\,(\mathrm{w}/Y) \subset [\mathrm{w}\,[Y]] \times [1, |\mathrm{w}\,[Y]\,|]$*) :* $count\,(\mathrm{w}/Y) = \{(v, count\,(\mathrm{w}\,[Y]\,/v))\,|v \in [\mathrm{w}\,[Y]]\}$.

10.9 Definíció (Kiterjesztett sor számlálója)
Legyen $Y = (Y_1, Y_2)$ *egy kijelölés és* w *duális mondat* $M\,(L)$ *felett és legyen* $t \in L$ *kiterjesztett sor. Az* Y *kijelölés* t *sorra vonatkoztatott számlálója* $count\,(t/Y) = count\,(\mathrm{w}/Y)$. *Legyenek* $t_1, t_2 \in L$, *a megfelelő duális mondatok* $\mathrm{w}_1, \mathrm{w}_2 \in D\,(L)$. *Azt mondjuk, hogy az* Y *kijelölés a* t_1 *és* t_2 *sorra vonatkoztatott számlálója egyenlő (jelölve* $t_1\,[Y] \stackrel{c}{=} t_2\,[Y]$*), ha* $count\,(t_1/Y) = count\,(t_2/Y)$.

10.10 Definíció (Reguláris funkcionális függőség számlálókkal)
Legyen L *reguláris nyelv és legyen* $M\,(L)$ *az* L *nyelvet elfogadó véges automata gráf reprezentációja. Legyen* $X = (X_1, X_2)$ *és* $Y = (Y_1, Y_2)$ *két kijelölés* $M\,(L)$ *felett. Az* $M\,(L)$ *felett számlálókkal értelmezett funkcionális függőségnek (reguláris FDc; RFDc) nevezzük az* $X \stackrel{c}{\rightarrow} Y$ *kifejezést. Az* $I \in \mathcal{I}\,(L)$ *(véges) adatbázis előfordulás kielégíti az* $X \stackrel{c}{\rightarrow} Y$ *funkcionális függőséget (jelölve* $I \stackrel{c}{\models} X \stackrel{c}{\rightarrow} Y$*), ha bármely két* $t_1, t_2 \in I$ *sorra* $t_1\,[X] \stackrel{c}{=} t_2\,[X]$ *csak akkor teljesülhet, ha* $t_1\,[Y] \stackrel{c}{=} t_2\,[Y]$ *is igaz.*

10.3 Megjegyzés (Számlálós RFD és RFD viszonya)
A számlálókon értelmezett RFDc az M2 Megkötés fenntartása esetén semmitmondó: miután egy előfordulás minden sorának azonos típusa (duális mondat) van, így bármely RFDc esetén a bal oldalak is és a jobb oldalak is azonosak lesznek. Az M2 Megkötés feloldásával nemtriviális eredményt kapunk. Nyilván, ha az L reguláris nyelv valamely két t_1, t_2 mondata gyengén (erősen) kielégíti az $X \to Y$ RFD-t akkor az $X \xrightarrow{c} Y$ is teljesül a két mondaton, de $L \overset{c}{\models} X \xrightarrow{c} Y$ nem biztos, hogy teljesül.

10.4. Példa. *A 7.3. Példában egy adott Előfizető-höz tartozó T_év elem a kapcsolat (telefonvonal) kiépítésének (aktiválásának) kezdő évét adja meg. Ettől az évtől kezdve a telefontársaság a vonalat évente ellenőrzi. Minden ellenőrzési eseményhez, a V_év elem tartalmazza az ellenőrzés év adatát, a V_kód elem pedig egy speciális kódot amely a T_év elem értékétől és az ellenőrzés évétől (V_év elem) függ. A példában adott RFD_{1a} függőség a létrehozás éve és az ellenőrzési adatokra elvárt épségi megszorítást írta le. A számlálók módszerét használva leírhatjuk azt a megszorítást, hogy az ellenőrzések év adatai és a megfelelő kódadatok száma azonos kell, hogy legyen:*

$$RFDc_{1a} : (\{\}, (\{V\_év\}, \{\})) \xrightarrow{c} (\{\}, (\{V\_kód\}, \{()\}))$$

Az XML dokumentumban realizálódó adatbázis előfordulás kielégíti $RFDc_{1a}$-t:

$\mathrm{w}_1 = \lfloor Név \mid T\_év \mid V\_év \mid V\_kód \mid V\_év \mid V\_kód \mid Dsl \mid Iptv \mid Díj \rceil$
$t_1 = \lfloor Kiss \mid 2008 \mid 2009 \mid 17 \mid 2010 \mid 21 \mid Családi \mid Alap \mid 30 \rceil$
$\mathrm{w}_1[X] = \lfloor V\_év \mid V\_év \rceil, \mathrm{w}_1[Y] = \lfloor V\_kód \mid V\_kód \rceil$
$\mathrm{w}_2 = \lfloor Név \mid T\_év \mid Dsl \mid Iptv \mid Díj \rceil$
$\mathrm{w}_2[X] = \epsilon, \mathrm{w}_2[Y] = \epsilon$
$t_2 = \lfloor Nagy \mid 2011 \mid Családi \mid Alap \mid 25 \rceil$
$\mathrm{w}_3 =$
$= \lfloor Név \mid T\_év \mid V\_év \mid V\_kód \mid V\_év \mid V\_kód \mid Dsl \mid Iptv \mid Telefon \mid Díj \rceil$
$t_3 = \lfloor Jeney \mid 2008 \mid 2009 \mid 17 \mid 2010 \mid 21 \mid Családi \mid Családi \mid Felező \mid 50 \rceil$
$\mathrm{w}_3[X] = \lfloor V\_év \mid V\_év \rceil, \mathrm{w}_3[Y] = \lfloor V\_kód \mid V\_kód \rceil$

mert két olyan sor van, a Kiss és Jeney előfizetők adatai, ahol a függőség bal oldala azonos gyakorisággal tartalmazza a kijelölt attribútumokat, és itt a jobb oldalakra is ez teljesül, azaz $t_1[X] \overset{c}{=} t_3[X]$ ($count(t_1/X) = count(t_3/X) = \{(V\_év,2)\}$) és $t_1[Y] \overset{c}{=} t_3[Y]$ ($count(t_1/Y) = count(t_3/Y) = \{(V\_kód,2)\}$).

Érdekes eredményre vezet egy vegyes, érték+számláló modell, ha a bal oldalon az RFD szokásos vetítését (értékazonosság) használjuk, míg a jobb oldalon számlálókat alkalmazunk: a módosított RFDc-t kielégíti egy előfordulás,

ha bármely két sorra $t_1[X] = t_2[X]$ *esetén* $t_1[Y] \overset{c}{=} t_2[Y]$ *is teljesül. Értelmezzük a*

$$RFDc_{1b} : (\{T\_év\}, (\{\}, \{\})) \overset{c}{\rightarrow} (\{\}, (\{V\_év, V\_kód\}, \{(V\_év, V\_kód)\}))$$

függőséget ezen vegyes modell szerint (ha a telepítés éve azonos két sorban, akkor az ellenőrzések száma is azonos). Az előfordulás ezt a függőséget is kielégíti, mert

$t_1[X] = 2008, t_1[Y] = \lfloor V\_év \mid V\_kód \mid V\_év \mid V\_kód \rceil$
$t_2[X] = 2011, t_2[Y] = \epsilon$
$t_3[X] = 2008, t_3[Y] = \lfloor V\_év \mid V\_kód \mid V\_év \mid V\_kód \rceil$

így $2008 = t_1[X] = t_3[X]$ *és* $t_1[Y] \overset{c}{=} t_3[X]$ *(*$= \{(V\_év,2), (V\_kód,2)\}$*).*

10.1 Propozíció (Számlálós RFDc implikációja)
A 6.1. Algoritmus a számlálós RFD (RFDc) implikációját is helyesen dönti el.

Bizonyítás
A 6.1. Algoritmus lépéseiben az „színezés”-t értelmezzük a két gráf csúcspontjainak a bejárás során történő azonos, illetve különböző számú érintéseként (azonos, illetve különböző színek). Ezzel az értelmezéssel az RFDc logikai implikációját helyesen dönti el a 6.1. Algoritmus ha az implikáció teljesül. Ha viszont az algoritmus végén az ellenpélda egy olyan előfordulást alkot, amelynek két sora a függőséghalmaz valamennyi függőségét kielégíti az RFDc kiértékelés szerint, de az implikálandó függőséget nem, akkor az implikálandó függőség jobb oldalának legalább egy attribútumát az algoritmus lépései során nem érintettük. Ez viszont azt jelenti, hogy az implikáció RFDc kiértékelés szerint sem teljesül, azaz a 6.1. Algoritmus az RFDc-k logikai implikációját is helyesen dönti el.

10.5. RFD megszámlálható szabályhalmazzal

A reguláris nyelvet definiáló grammatika helyettesítési szabályainak halmazát végesnek adtuk meg. A nemterminális szimbólumok végtelen számosságára nincs természetes igény, a terminális szimbólumok, mint adatok, lehetnek végtelen számosságúak: például természetes számok. Meg kell vizsgálnunk az RFD modellt megszámlálhatóan végtelen számosságú szabályhalmaz, azaz megszámlálható végtelen sok terminális szimbólum esetén is. Miután ebben a fejezetben gyakran hivatkozunk megszámlálhatóan végtelen halmazokra,

hacsak nem jelezzük másként, a „végtelen", „megszámlálható" kifejezéseken „megszámlálhatóan végtelen"-t értünk.

A szakirodalomban több modellt fogalmaztak meg reguláris nyelvek végtelen ábécére való kiterjesztésére. Az első, általunk ismert megoldás Autebert és munkatársaitól származik [3, 9]. Többek között definiálták a H-racionális nyelv fogalmát: a D végtelen ábécén értelmezett L nyelv, amelyre minden X véges ábécé esetén minden $\Phi : D^* \to X^*$ alfabetikus homomorfizmusra a $\Phi(L)$ nyelv reguláris. Ez a modell érdekes tulajdonságokkal rendelkezik (minden H-racionális nyelv egyúttal adatnyelv is, de a fordítottja nem igaz), de az RFD modell kiterjesztésére nem használható, mert egy reguláris nyelvhez nem lehet megtalálni a megfelelő H-racionális párt.
A végtelen ábécék kezelésére egy másik módszert, a véges ábécével való kódolást is kidolgozták. Az eredeti, végtelen ábécét egy véges ábécével kódolják, és azután a kódszavak alkotta nyelvvel (amely véges ábécén alapszik) dolgoznak. Példa erre [45, 15], ahol is a $\Sigma = \{a_i | i \geq 1\}$ végtelen ábécé a_i szimbólumát a 0^i1 kódszó kódolja a $\{0,1\}$ kétbetűs (véges) ábécé felett. A Σ végtelen ábécé felett egy nyelvet azután regulárisnak definiál, ha a megfelelő kódszavak alkotta nyelv a bináris $\{0,1\}$ ábécé felett, a véges ábécék értelmében reguláris. 1994-ben született a véges memóriájú automaták modellje (Kaminski és Francez [33], továbbfejlesztették [21, 34, 43]). A véges automata kiegészül a végtelen ábécé betűit tárolni képes véges sok regiszterrel. Az automata átmeneteit egy adott állapotból a beolvasott következő betűn kívül a regiszterek aktuális állapota is befolyásolja (a regiszterek tartalma a bejövő jelekkel és egymással összehasonlítható, a bejövő jelek tárolhatók, ilyenkor a regiszterben tárolt jel felülíródik). A felismert nyelvek számos érdekes tulajdonsággal (unióra, metszetre zárt) rendelkeznek, és ha Γ egy véges részhalmaza a D végtelen bejövő ábécének és L az elfogadott nyelv, akkor $L \cap \Gamma^*$ reguláris. A véges memóriájú automatákat regiszter automatáknak is nevezik.

A regiszter automaták szemléltetésére nézzük meg a 2.1. Példát (relációs adatbázis reguláris grammatikával megadott reguláris nyelvként generálva) végtelen ábécén. A regiszter automaták végtelen ábécén a regiszterekben az induló állapotban tárolt betűkből álló véges ábécén értelmezett reguláris nyelvet ismerik fel ([21]), ha

1. egy regiszter sem üres a kezdő állapotban
2. nincsenek definiálva a regisztereket író utasítások

10.5. Példa. *Visszautalva a 2.1. Példára, legyen R(A,B,C,D,E) egy relációs séma, úgy, hogy*

A,B,C,D,E az attribútumok halmaza,
***dom** jelöli az attribútumok közös értéktartományát (az attribútumokhoz különböző értéktartományokat is rendelhetünk).*

Legyen továbbá G=(N,T,S,P) reguláris grammatika, ahol

$N = \{R, A, B, C, D, E, \mathcal{ENDE}\}$ *a nemterminális szimbólumok halmaza,*
$T = \{\langle, \rangle\} \cup dom$ *a terminális szimbólumok (értékek) halmaza,*
$S = R$ *a kezdő (start) szimbólum,*
$P =$
$= \{R \Rightarrow \langle\, A, A \Rightarrow \mathbf{a}B, B \Rightarrow \mathbf{b}C, C \Rightarrow \mathbf{c}D, D \Rightarrow \mathbf{d}E, E \Rightarrow \mathbf{e}\,\mathcal{ENDE}, \mathcal{ENDE} \Rightarrow\rangle\}$*;*
(ahol $\mathbf{a}, \mathbf{b} \ldots \subset dom$*) a helyettesítési szabályok (szabálysémák).*

Az L(G) nyelv, amelyet a G grammatika generál, $\langle abcde\rangle\,, a{\in}\mathbf{a}, b{\in}\mathbf{b}, \ldots, e{\in} \in \mathbf{e}$ *sorozatokból áll, minden ilyen sorozat („sor") a **dom** értéktartományból vett értékek (szimbólumok) összefűzéseként (konkatenáció) áll elő, és az R reláció minden előfordulása az L(G) részhalmaza (L(G) véges!).*
Legyen dom végtelen halmaz és legyenek $R_A, R_B, R_C, R_D, R_E \subset dom$ *páronként diszjunkt, véges részhalmazok. Legyen* $r{=}|R_A|{+}|R_B|{+}|R_C|{+}|R_D|{+}|R_E|$*. Definiáljunk egy A regiszter automatát, amelynek* $r{+}2$ *számú regisztere van, amelyekből* $|R_A|$ *számú regiszterben az* R_A *halmaz értékei vannak kezdőértékként (mindegyik regiszterben más érték, a hozzárendelés tetszőleges), majd* $|R_B|$ *számú regiszterben az* R_B *halmaz értékei és így tovább. (Két speciális regiszter legyen* $r_R{=}\langle$ *és* $r_{\mathcal{ENDE}}{=}\rangle$ *az adott kezdőértékkel). Az automata ne írja a regisztereket, átmenetdefiníciói* (p, reg, q) *alakúak, ami azt jelenti, hogy ha a p állapotban beolvasott jel a reg regiszter tartalmával egyenlő, akkor az automata átmehet a q állapotba. Legyen* q_R *az automata kezdőállapota, és legyen* (q_R, reg_R, q_A)*, valamint* $(q_E, reg_{\mathcal{ENDE}}, q_{\mathcal{ENDE}})$*, ahol*$q_{\mathcal{ENDE}}$ *az automata elfogadó állapota. Legyenek* $\left(q_A, reg_{A(i)}, q_B\right); i \leq |R_A|$ *azon átmenetdefiníciók, amelyek az* R_A *halmaz elemeit tartalmazó regiszterekre hivatkoznak. (A további átmenetdefiníciók hasonlóan alakuljanak). Ez az automata nyilván elfogadja az L(G) nyelv szavait, míg minden* $dom \setminus (R_A \cup R_B \cup R_C \cup R_D \cup R_E)$*-beli szimbólumra leáll.*

A regiszter automaták egy speciális esete az adatszavakon értelmezett nyelveket ismeri fel. Legyen Γ véges, Σ végtelen ábécé, akkor a $\Gamma \times \Sigma$ ábécén értelmezett szavakat nevezik adatszavaknak (ez a modell jól alkalmazható XML komponensek leírására, ha Γ a *tag* jelölőket, Σ az XML szövegértékeket reprezentálja). Az adatszavak nyelvhez, a reguláris nyelvekhez hasonlóan,

speciális reguláris kifejezések is társíthatóak (reguláris kifejezések memóriával [37]).

A fenti modellek végtelen ábécéket kezelnek és reguláris nyelvekhez hasonlatosak (általában véges metszetük reguláris nyelv), de a szavak elfogadásának kritériumai a szavat alkotó szimbólumok közötti egyenlőség (egyenlőtlenség) feltétele, amely feltételeket az írható/olvasható regiszterek segítségével ellenőriznek. A mi modellünk végtelenre való kiterjesztésére intuitíve arra lenne szükség, hogy az automata állapotaiban (véges sok) elfogadható szimbólumokat (végtelen sok) valamilyen kiszámítható függvénnyel eldönthessük. Ebben a fenti modellek nem segíthetnek. Egy újabb modell némi átalakítással a mi esetünkre is használható, erre később rátérünk (10.14. és 10.16. Definíciók).
Előbb azonban szükségünk lesz néhány új fogalomra.

Az 1. Fejezetben már informálisan bevezettük a „szabályséma" fogalmát, erre ott csak azért volt szükség, hogy a nagyszámú, de véges sok szabályra (amelyek ugyanazt a két nemterminális szimbólumot, vagy a véges automata ugyanazon két állapotát kapcsolták össze) összefoglalóan hivatkozhassunk. A végtelen ábécére való kiterjesztés céljából most formális definíciót adunk erre a fogalomra.

10.11 Definíció (Szabályséma)
Legyen $G = (N, T, S, P)$ *reguláris grammatika úgy, hogy valamely* $R_{X,Y} \subset$ $\subset T; X, Y \in N$ *részhalmazra teljesül, hogy* $\forall x \in R_{X,Y}$*-re igaz, hogy* $X \Rightarrow xY \in P$. *Ekkor az* $X \Rightarrow Y$ *kifejezést szabálysémának nevezzük és az* $\{X \Rightarrow xY | x \in R_{X,Y}\}$ *szabályhalmazzal azonos jelentéssel használjuk.*

10.4 Megjegyzés
Az $M = (Q, \Gamma, \delta, S, \mathcal{F})$ *véges automata adjacencia mátrixát* $A(M) = (a_{p,q})$*-val jelöljük* $(p, q \in Q)$. $a_{p,q} > 0$ *akkor és csak akkor, ha* $\exists x \in \Gamma, \delta(p, x) = q$.

10.12 Definíció (Átmenetséma véges automatán)
Legyen $M = (Q, \Gamma, \delta, S, \mathcal{F})$ *véges automata úgy, hogy valamely* $p, q \in Q$ *állapotokra* $\exists x' \in \Gamma$ *úgy, hogy* $\delta(p, x') = q$.
Legyen $p, q \in Q$ *és* $R_{p,q} = \{x'' | x'' \in \Gamma; \delta(p, x'') = q\}$. *Ekkor a* $\delta(p) = q$ *kifejezést átmenetsémának és a* $\hat{\delta} = \{\delta(p) = q | a_{p,q} > 0\}$ *halmazt az* M *automata átmenetséma halmazának nevezzük.*
A $\delta(p) = q$ *átmenetsémát a* $\{\delta(p, x') = q | x' \in R_{p,q}\}$ *átmenetdefiníciókkal azonos jelentéssel használjuk. A* $\delta(p) = q$ *átmenetséma pontosan azokra a* $p, q \in Q$ *állapotokra definiált amelyekre az* M *automata adjacencia mátrixának megfelelő eleme* > 0. *A* $\Delta(M) = \{(p, q) \mid (p, q) \in Q \times Q; a_{p,q} > 0\}$ *halmazt az* M

véges automata átmenetséma indexhalmazának nevezzük.
Az $R_{p,q}=\{x'|x'\in\Gamma;\delta(p,x')=q;(p,q)\in\Delta(M)\}$ *halmazokat (az automatát a p állapotból a q állapotba átvivő szimbólumok halmazait) átmenethalmazoknak nevezzük.*

Az alábbi lemma a 10.4. Megjegyzésből következik.

Legyen $M=(Q,\Gamma,\delta,S,\mathcal{F})$ véges automata, legyen $A=A(M)$ és $K=\max_{p,q}(a_{p,q})$.

Az M automata átmenetséma halmaza legyen $\hat{\delta}=\{\delta(p)=q|(p,q)\in\Delta\}$, legyen $\Delta=\Delta(M)$ az M átmenetséma indexhalmaza.

10.1 Lemma
Legyenek $R_{p,q}=\{x'|x'\in\Gamma;\delta(p,x')=q;(p,q)\in\Delta\}$ *az* M *véges automata átmenethalmazai. Akkor* $\Gamma=\bigcup_{p,q\in\Delta}R_{p,q}$.

A fenti definíciók segítségével könnyen jellemezhetjük a determinisztikus véges automatákat:

10.2 Lemma
Az $M=(Q,\Gamma,\delta,S,\mathcal{F})$ *véges automata pontosan akkor determinisztikus, ha átmenethalmazaira teljesül, hogy ha* $p,q_1,q_2\in Q$ *és* $q_1\neq q_2$ *akkor* $R_{p,q_1}\cap R_{p,q_2}=\emptyset$.

10.13 Definíció (Átmenetséma ekvivalencia)
Legyen $M=(Q,\Gamma,\delta,S,\mathcal{F})$ *determinisztikus véges automata, legyen* $L=L(M)$ *az* M *által felismert reguláris nyelv. Legyen* $\mathrm{w}_1,\mathrm{w}_2\in L$ *két szó, amelyekre teljesül, hogy* $|\mathrm{w}_1|=|\mathrm{w}_2|=n>0$ *és legyen* $\mathrm{w}_i=v_i(1)\ v_i(2)\dots\ v_i(n)\,;i=1{,}2$ *úgy, hogy* $\forall k\in[1,n]\ \exists(p,q)\in\Delta;v_i(k)\in R_{p,q};i=1{,}2$. *Akkor azt mondjuk, hogy* w_1 *és* w_2 *átmenetséma ekvivalensek (jelölve* $\mathrm{w}_1\equiv_{\hat{\delta}}\mathrm{w}_2$*).*

10.3 Lemma
Az átmenetséma ekvivalencia ekvivalencia reláció.

Legyen $M_\Gamma=(Q,\Gamma,\delta,S,\mathcal{F})$ véges automata, D végtelen ábécé úgy, hogy $\Gamma=\bigcup_{p,q\in Q}D_{p,q};D_{p,q}\subset D;\forall x'\in D_{p,q};\delta(p,x')=q$ (Γ véges, így minden $D_{p,q}$ is az).

Ekkor az átmenetfüggvény a $\hat{\delta}=\{\delta(p)=q|p,q\in Q;D_{p,q}\neq\emptyset\}$ átmenetséma halmazzal adható meg.

Legyen $L(M_\Gamma)$ az M_Γ által elfogadott reguláris nyelv.

10.1. Állítás. *Az* $L=\bigcup_{\Gamma\subset_{fin}D}L(M_\Gamma)$ *(ahol* $\subset_{fin}$ *véges részhalmazt jelent) nyelv nem reguláris.*

Bizonyítás
Legyen $\mathrm{w} \in L$ *nem üres szó, akkor van olyan* $\Gamma_0 \subset_{fin} D$ *amelyre* $\mathrm{w} \in L(M_{\Gamma_0})$. *Legyen* $\mathrm{w}{=}v_1 v_2 \ldots v_n$, *és legyen* $\Gamma{=}D \backslash \Gamma_0{=}\{u_1, u_2, \ldots\}$. *A* w *szó elfogadásakor az* M_{Γ_0} *automata az utolsó lépésben valamely* q *állapotból lépett a (mondjuk)* q_{end} *elfogadó állapotba, azaz* $\delta(p, v_n) = q_{end}$, *vagyis* $v_n \in R^{\Gamma_0}_{q,q_{end}}$. *Legyenek* $\Gamma_i{=}\Gamma_{i-1} \cup \{u_i\}\,; i \geq 1$ *és* $R^{\Gamma_i}_{q,q_{end}} = R^{\Gamma_{i-1}}_{q,q_{end}} \cup \{u_i\}$, *és* $R^{\Gamma_i}_{p_1,p_2} = R^{\Gamma_0}_{p_1,p_2}$, *ha* $(p_1, p_2) \neq$ $= (q, q_{end})$. *Akkor* $\mathrm{w}_i = v_1\, v_2\, \ldots\, v_{n-1}\, u_i \in L(M_{\Gamma_i})\,; i \geq 1$, *azaz* $\mathrm{w}_i \in L; i \geq 1$. *De ez azt jelenti, hogy az* L*-et elfogadó véges automatának (amely létezik, ha a feltevés szerint* L *reguláris) az elfogadó állapotához végtelen sok átmenet kellene, hogy vezessen.*

A megszámlálható ábécén működő, variálható (változókat használó) véges automatákat (variable finite automata over infinite alphabets - VFA) Grumberg et al. vezették be [23].

10.14 Definíció (Variálható véges automata [23])
Legyen $M = (Q, \Gamma, \delta, S, \mathcal{F})$ *véges automata,* D *végtelen ábécé úgy, hogy* $\Gamma =$ $= \Sigma \cup X \cup \{y\}$ *(ahol* Σ *véges ábécé,* $\Sigma \subset D$, X *kötött változók véges halmaza és* y *szabad változó) akkor a* $V = (D, M)$ *párost variálható (változókat használó) véges automatának (VFA) nevezzük. Az* M *véges automatát a* V *variálható véges automata mintaautomatájának nevezzük.*

Legyen $\mathrm{w} \in L(M)$ a $V = (D, M)$ variálható véges automata M mintaautomatája által felismert szó. Legyen $\mathrm{w} = \mathrm{w}_1 \mathrm{w}_2 \ldots \mathrm{w}_n$ és legyen $v \in D^*$ úgy, hogy $v{=}v_1 v_2 \ldots v_n$. Azt mondjuk, hogy v a w érvényes előfordulása, ha $\forall i, j \in [1, n]$ az alábbi három feltétel valamelyike teljesül:

1. $v_i = \mathrm{w}_i$ és $\mathrm{w}_i \in \Sigma$
2. ha $v_i, v_j \in X$, úgy $\mathrm{w}_i = \mathrm{w}_j$ akkor és csak akkor teljesül, ha $v_i = v_j$ és $\mathrm{w}_i, \mathrm{w}_j \notin \Sigma$
3. ha $v_i = y$ és $v_j \neq y$ akkor $\mathrm{w}_i \neq \mathrm{w}_j$.

A $V(D, M)$ variálható véges automata által felismert nyelvnek nevezzük, és $L(V)$-vel az $L(A)$ reguláris nyelv szavaihoz tartozó érvényes előfordulások halmazát [23]. (Nyilván, $L(V) \subseteq D^*$). Ezek a nyelvek általában nem regulárisak, de számos, a reguláris nyelvekre hasonlító tulajdonságuk van (Zártak unióra és metszetre, de nem zártak a komplementerképzésre [23]).

10.15 Definíció (DFA megadott átmenethalmazokon)
Legyen $M=(Q,\Gamma,\delta,S,\mathcal{F})$ *determinisztikus véges automata, legyen* $\Delta=\Delta(M)$ *az* M *átmenetséma indexhalmaza. Legyenek az* M *átmenethalmazai páronként diszjunktak, azaz* $R_{p_1,q_1}\cap R_{p_2,q_2}=\emptyset;(p_1,q_1)\neq(p_2,q_2)$.
Legyen $K=\max\limits_{(p,q)\in\Delta}(|R_{p,q}|)$. *Legyen* D *végtelen ábécé és legyenek* $D_{p,q}\subset D;(p,q)\in$ $\in\Delta$ *páronként diszjunkt részhalmazok (átmenethalmazok) úgy, hogy* $0<|D_{p,q}|<$ $<K;(p,q)\in\Delta$. *Jelöljük* $D_\Delta=\{D_{p,q}|\,(p,q)\in\Delta\}$*-val a kiválasztott átmenethalmazok halmazát.* D_Δ *alapján megadhatunk egy, az* M*-ből származtatott véges automatát:* $M(D_\Delta)=(Q,\Sigma,\delta',S,\mathcal{F})$, *ahol* $\Sigma=\bigcup\limits_{p,q\in\Delta}D_{p,q}$, $\delta'(p,x')=q$ *akkor és csak akkor, ha* $x'\in D_{p,q}$.

Az $M(D_\Delta)$ *véges automatát az* M *véges automata* D_Δ *halmazon felvett értékének nevezzük.*

10.16 Definíció (Korlátos variálható véges automata)
Legyen $M=(Q,\Gamma,\delta,S,\mathcal{F})$ *véges automata, legyen* $\Delta=\Delta(M)$ *az* M *átmenetséma indexhalmaza. Legyenek az* M *állapothalmazai páronként diszjunktak, azaz* $R_{p_1,q_1}\cup R_{p_2,q_2}=\emptyset;(p_1,q_1)\neq(p_2,q_2)$. *Legyen* $K=\max\limits_{(p,q)\in\Delta}(|R_{p,q}|)$. *Legyenek* $X_{p,q}=\{x_{p,q}(1),x_{p,q}(2),\ldots,x_{p,q}(K);(p,q)\in\Delta\}$ *kötött változók páronként diszjunkt véges halmazai.*

Legyen $X_\Delta=\{X_{p,q}|X_{p,q};(p,q)\in\Delta\}$ *a kötött változók halmazainak halmaza. Legyen* $\Sigma=\bigcup\limits_{p,q\in\Delta}X_{p,q}$ *és legyen* $M'=M(X_\Delta)$ *az* M *véges automata a kötött változókon, mint átmeneteken felvett értéke. Legyen* D *végtelen ábécé, akkor a* $V=(D,M',K)$ *hármast korlátos variálható (változókat használó) véges automatának (limited VFA,LVFA) nevezzük. Az* M *véges automatát a* V *korlátos variálható véges automata mintaautomatájának nevezzük.*

Legyen $M_\Gamma=(Q,\Gamma,\delta,S,\mathcal{F})$ véges automata, legyen $A=A(M)$ és $K=$ $=\max\limits_{p,q\in Q}(a_{p,q})$.
Az M automata átmenetséma halmaza legyen $\hat{\delta}=\{\delta(p)=q|\,(p,q)\in\Delta\}$, ahol $\Delta=\Delta(M)$ az M átmenetséma indexhalmaza. Legyen D végtelen ábécé és legyen $S_{D,\Delta,K}=\{D_{p,q}|\,(p,q)\in\Delta;D_{p,q}\subset D;0<|D_{p,q}|\leq K\}$ a D részhalmazainak halmazából egy kiválasztás. Legyen $\Lambda(S_{D,\Delta,K})=\bigcup\limits_{(p,q)\in\Delta}D_{p,q};D_{p,q}\subset$ $\subset D;0<|D_{p,q}|\leq K$; ($\Lambda(S_{D,\Delta,K})$ véges).

Legyen $L\left(M_{\Lambda\left(S_{D,\Delta,K}\right)}\right)$ az $M_{\Lambda\left(S_{D,\Delta,K}\right)}$ által elfogadott reguláris nyelv.
Legyen $V=(D,M',K)$ az M mintaautomatához a 10.16 definíció szerint rendelt korlátos variálható véges automata.

10.2. Állítás. *Az* $L=\bigcup\limits_{S_{D,\Delta,K}}L\left(M_{\Lambda\left(S_{D,\Delta,K}\right)}\right)$ *nyelvre teljesül, hogy* $L\subseteq L(V)$.

Bizonyítás
Minden w$\in L$*-re van olyan* $S_{D,\Delta,K}=\{D_{p,q}|\,(p,q)\in\Delta; D_{p,q}\subset D; 0<|D_{p,q}|\leq K\}$ *halmaz, hogy az* M_Γ *automata* $S_{D,\Delta,K}$*-n felvett értéke elfogadja* w*-t. Miután a* V *definiálásához felhasznált* M' *automata az* M_Γ *automata* X_Δ *halmazon felvett értéke, és* $|X_{p,q}|=K; (p,q)\in\Delta$*, így a* w *szó valamely* $v\in L\,(M')$ *érvényes előfordulása.*

11. FD környezetfüggetlen nyelveken

Az XML dokumentumok számára elsőként bevezetett sémanyelv, a DTD elemleíró formalizmusa lényegében kiterjesztett környezetfüggetlen grammatikának (extended context free grammar - $ECFG$) felel meg ([40]), mivel $a \Rightarrow r$ formájú szabályokat tartalmaz, ahol a egy elem neve és r az elemnevek (mint szimbólumok!) ábécéje feletti reguláris kifejezés. Megjegyezzük, bár erre a tényre nem lesz szükségünk, hogy az $ECFG$-k pontosan a környezetfüggetlen nyelveket generálják.
A reguláris nyelveken értelmezett funkcionális függőség, mint láttuk, nem fedi le az XML adatbázisokban előforduló eseteket, de a modell egy természetes általánosításával a (kiterjesztett) környezetfüggetlen nyelvekre, azaz a „horizontális" reguláris kifejezéseket „vertikális" irányban összekapcsolva konstruálhatunk egy összetettebb adatmodellt amely már alkalmas lesz az XML világban felmerülő összetettebb megszorítások leírására is.

A következőkben általánosítjuk a reguláris funkcionális függőség definícióját kiterjesztett környezetfüggetlen grammatika által generált nyelvekre, azaz a környezetfüggetlen nyelvekre. Az $ECFG$ egy $G = (N, T, S, P)$ négyes, ahol

N a nemterminális szimbólumok (véges) halmaza,
T a terminális szimbólumok (véges) halmaza, úgy, hogy $N \cap T = \emptyset$,
P az $A \Rightarrow R_A$ formájú levezetési szabályok halmaza, ahol $A \in N$, és R_A reguláris kifejezés $N \cup T$ felett,
$S \in N$ a kezdő szimbólum.

11.1 Megjegyzés
Az XML sémanyelveket generáló $ECFG$-k helyettesítési szabályainak jobboldalán nemterminális szimbólumok reguláris kifejezései állnak, mivel a sémanyelvek csak az elemnevekből képzett lehetséges sorozatokat - adatbázis terminológiával, a lehetséges sémákat - írják le. A számunkra szükséges modellben adatbázis sorokat (XML elemeket) szeretnénk előállítani ezért a reguláris kifejezésekben terminális szimbólumok - „értékek" - is szerepelnek. Az $ECFG$ modellt a DTD sémanyelv sugallja, de nem a DTD modellezése a célunk, hanem a reguláris nyelvekre épített függőségeknél erősebb kifejező erejű függőségek leírása.

A következőkben az $ECFG$-t leszűkített értelemben használjuk, csak a levezetési szabályok két, alább következő formáját engedjük meg:

1. $A \Rightarrow R_A$, ahol $A \in N$, R_A reguláris kifejezés N felett,

2. $A \Rightarrow u$, ahol $A \in N$, $u \in T$.

Feltesszük továbbá, hogy a két csoport baloldali nemterminálisai diszjunkt halmazokat alkotnak. Az két csoport baloldali nemterminálisait 1. és 2. típusú nemterminálisoknak nevezzük, legyenek ezek halmazai N_1, N_2, akkor igaz, hogy $N = N_1 \cup N_2; N_1 \cap N_2 = \emptyset$.

11.2 Megjegyzés
Az ECFG levezetési szabályait azért korlátozzuk a fenti két alakra, mert a terminális szimbólumokat össze szeretnénk kapcsolni típusként szereplő nemterminális szimbólumokkal, hasonlóan a reguláris nyelvnél alkalmazott módszerhez, ahol is a nemterminális szimbólumot az őt helyettesítő terminális szimbólum(halmaz) típusaként interpretáltuk. Az ilymódon szűkített grammatikát ECFGR-nek nevezzük.

11.1. Példa. *Ez a példa az ECFG szűkítésének hatását mutatja meg azonos adattartalmú, de különböző formátumú XML dokumentumokon. Az XML-séma mixed=„true" opcióját (jelentése: az összetett típus beágyazott elemeket és szabad szövegeket is tartalmazhat) felhasználva, illetve azt elhagyva különböző reguláris kifejezéseket kapunk: a vegyes típusnál (mixed=„true") az eredeti ECFG értelmezésnek megfelelő, terminálisokat és nemterminálisokat is tartalmazó reguláris kifejezés áll a jobb oldalon, míg az opció elhagyása esetén (jelentése: elements only) a reguláris kifejezés tisztán nemterminálisokból áll. A példa két azonos adattartalmú, de különböző formátumú XML dokumentumot mutat, az értelemszerűen különböző XSD-kel leírt Sorozatlevél adatbázist. Nézzük először a vegyes adatokat tartalmazó típus XML-séma leírását:*

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
       elementFormDefault="qualified">
<xs:element name="Sorozatlevél">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Levél" type="t_Levél"
                minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="t_Levél" mixed="true">
    <xs:sequence>
```

```
      <xs:element name="Név" type="xs:string"/>
      <xs:element name="Lakcím" type="xs:string"/>
      <xs:element name="Szöveg" type="xs:string"/>
  <xs:element name="Dátum" type="xs:date"/>
      <xs:element name="Aláírás" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A sémaleírás szerint készült dokumentum Levél eleme a

$R_{Levél}$=„Kedves”+„Tisztelt” ∘ „Dr.”? ∘ <Név> ∘ (1125 + 9024 + ...) ∘ <Lakcím> ∘ <Szöveg> ∘ <Dátum> ∘ „Dr.”? ∘ <Aláírás>

reguláris kifejezésnek felel meg, azaz a Sorozatlevél adatbázis ECFG grammatikája tartalmazza a Levél $\Rightarrow R_{Levél}$ *helyettesítési szabályt. A sémának (a reguláris kifejezésnek) megfelelő előfordulás az alábbi XML dokumentum:*

```
<?xml version="1.0"?>
<Sorozatlevél xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="file:letter.xsd">
<Levél>
    Kedves<Név>Kovács János</Név>
    1125<Lakcím>Budapest, Hal u. 1.</Lakcím>
    <Szöveg>Felszólítom ...</Szöveg>
    <Dátum>2014-01-21</Dátum>
    Dr.<Aláírás>Nagy Jakab</Aláírás>
</Levél>
<Levél>
    Tisztelt Dr.<Név>Török Péter</Név>
    9024<Lakcím>Győr, Gomba u. 6.</Lakcím>
    <Szöveg>Kérem ...</Szöveg>
    <Dátum>2014-01-20</Dátum>
    <Aláírás>Pók Éva</Aláírás>
</Levél>
</Sorozatlevél>
```

Elhagyva a sémából a mixed=„true” opciót, az alábbi sémával (és a megfelelő reguáris kifejezéssel) írhatjuk le az azonos adattartalmú, de a céljainknak jobban megfelelő, szűkített ECFG grammatikájú SorozatlevélE adatbázist:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                        elementFormDefault="qualified">
           <xs:element name="SorozatlevélE">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="LevélE" type="t_LevélE"
           minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="t_LevélE">
    <xs:sequence>
  <xs:element name="Megszólítás" type="xs:string"/>
      <xs:element name="Cím" type="xs:string"
           minOccurs="0" maxOccurs="1"/>
      <xs:element name="Név" type="xs:string"/>
  <xs:element name="Irányítószám" type="xs:integer"/>
      <xs:element name="Lakcím" type="xs:string"/>
      <xs:element name="Szöveg" type="xs:string"/>
  <xs:element name="Dátum" type="xs:date"/>
      <xs:element name="Cím" type="xs:string"
          minOccurs="0" maxOccurs="1"/>
      <xs:element name="Aláírás" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A módosított sémaleírás szerint készült dokumentum LevélE eleme a

$R_{LevélE}$=<*Megszólítás*> ∘ <*Cím*>? ∘ <*Név*> ∘ <*Irányítószám*> ∘ <*Lakcím*> ∘ <*Szöveg*> ∘ <*Dátum*> ∘ <*Cím*>? ∘ <*Aláírás*>

reguláris kifejezésnek felel meg, azaz a SorozatlevélE adatbázis szűkített ECFG grammatikája tartalmazza a $LevélE \Rightarrow R_{LevélE}$*, tisztán nemterminálisokból álló helyettesítési szabályt. A sémának (a reguláris kifejezésnek) megfelelő előfordulás az alábbi XML dokumentum:*

```
<?xml version="1.0"?>
<SorozatlevélE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="file:letter_e.xsd">
```

```
<LevélE>
    <Megszólítás>Kedves</Megszólítás>
    <Név>Kovács János</Név>
    <Irányítószám>1125</Irányítószám>
    <Lakcím>Budapest, Hal u. 1.</Lakcím>
    <Szöveg>Felszólítom ...</Szöveg>
    <Dátum>2014-01-21</Dátum>
    <Cím>Dr.</Cím><Aláírás>Nagy Jakab</Aláírás>
</LevélE>
<LevélE>
    <Megszólítás>Tisztelt</Megszólítás>
    <Cím>Dr.</Cím><Név>Török Péter</Név>
    <Irányítószám>9024</Irányítószám>
    <Lakcím>Győr, Gomba u. 6.</Lakcím>
    <Szöveg>Kérem ...</Szöveg>
    <Dátum>2014-01-20</Dátum>
    <Aláírás>Pók Éva</Aláírás>
</LevélE>
</SorozatlevélE>
```

Ez az XML dokumentum nem tartalmaz vegyes elemeket, azaz minden adatnak (szövegnek) van jelölője (elemneve). A SorozatlevélE és a LevélE jelölők 1. típusú, valamennyi többi jelölő (Megszólítás,Név,...) 2. típusú nemterminálisnak felelnek meg.

Az A nemterminális szimbólumból generált nyelv nem szükségképpen reguláris: jelöljük $L(A) \subseteq T^*$-val az A nemterminális szimbólumból a P-beli szabályokkal generált nyelvet. Ennek a nyelvnek a mondatait az A nemterminális szimbólum elemi összetett sorainak nevezzük.
Minden $A \Rightarrow R_A$ helyettesítési szabályban szereplő R_A reguláris kifejezés reguláris nyelvet ($L_A \subseteq N^*$) határoz meg, ezt a nyelvet elfogadó véges automata (jelöljük M_A-val) a 6.1. Algoritmussal konstruálható meg. A G grammatika szerinti levezetésekben egy nemterminális A szimbólumot vagy az L_A nyelv egy mondatával helyettesítjük (az 1. csoportba tartozó szabály esetén), vagy egy terminális szimbólummal (a 2. csoportba tartozó szabály végrehajtásakor).

11.3 Megjegyzés

Az általánosság megszorítása nélkül feltehetjük, hogy mindegyik nemterminális szimbólum pontosan egyszer szerepel valamely levezetési szabály bal oldalán, mert az $A \Rightarrow R_A^1$ és $A \Rightarrow R_A^2$ szabályokat helyettesíthetjük a $A \Rightarrow (R_A^1 + R_A^2)$

szabállyal úgy, hogy az így kapott grammatika az eredetivel ekvivalens ([4]). Feltehetjük továbbá, hogy egy levezetési láncban mindaddig 1. típusú helyettesítéseket végzünk, amíg vannak az aktuális mondatban 1. típusú nemterminálisok.

Legyen $\omega \in L(A)$, legyen $U \in N$ nemterminális szimbólum úgy, hogy $U \in \in R_A$. Az U nemterminális jelet az A nemterminális szimbólum egy atomi attribútumának nevezzük. U a $G_U = (N, T, U, P)$ $ECFGR$ grammatika kezdőszimbóluma, így amikor G a ω mondatot generálja, $L(U)$ mondatai közül is generálódnak némelyek, legyenek ezek a mondatok $u_1, \ldots, u_k$ $(k \geq 1)$. Akkor $\omega = \omega_1 u_1 \ldots \omega_k u_k \omega_{k+1}$, ahol $\omega_i \in T^*, 1 \leq i \leq k+1$. Az U atomi attribútum a t sorból a $t[U] = u_1 \circ u_2 \ldots \circ u_k$ jelsorozatot vetíti ki.

11.1 Definíció
Legyen $G = (N, T, S, P)$ egy $ECFGR$, legyen $\alpha \in N^*$ nemterminális jelek sorozata, akkor az $L(\alpha) \subseteq T^*$ nyelv az α-ból a P-beli helyettesítési szabályokkal levezethető jelsorozatok halmaza. Formálisan, legyen $\alpha = \alpha_1\alpha_2 \ldots \alpha_n, \alpha_i \in N, 1 \leq \leq i \leq n$, akkor $L(\alpha) = \{\omega \in T^* | \omega = \omega_1\omega_2 \ldots \omega_n\}, \alpha_i \Rightarrow^+ \omega_i$, ahol $\Rightarrow^+$ a levezetési reláció tranzitív lezártja.

11.1. Tétel. *Legyen $G = (N, T, S, P)$ egy $ECFGR$, legyen $A \in N$ és legyen $\alpha \in L_A$ és legyen $\omega \in L(\alpha)$. Legyen $\alpha = \{B_1B_2 \ldots B_n\}$ akkor $\omega = \{\omega_1\omega_2 \ldots \omega_n\}$ úgy, hogy $\omega_i \in L(B_i), 1 \leq i \leq n$.*

Bizonyítás
A levezetési lépések száma szerinti indukcióval látható, hogy az $\alpha \mapsto \alpha_1 \mapsto \alpha_2 \ldots \mapsto \alpha_m = \omega$ levezetési lánc mindegyik tagja n részsorozat összefűzéseként áll elő: $\beta_i \in (N \cup T)^*, 1 \leq i \leq n$, úgy, hogy β_i a B_i-ből levezetett jelsorozat. Ez nyilvánvalóan igaz az első lépésben. Ha a k-adik lépésben a β_i tag tartalmazza a Q nemterminális szimbólumot, amelyet a $k+1$-edik lépésben a $Q \Rightarrow R_Q$ szabály szerint kell helyettesíteni, azaz, $Q \Rightarrow Q_1Q_2 \ldots Q_s$, akkor legyen $\beta_i = = \gamma Q \eta$, valamely $\gamma\, \eta$ jelsorozatokkal ($\gamma\, \eta \in (N \cup T)^*$), ha $\beta_i \mapsto \beta_i'$ akkor $\beta_i' = = \gamma\, Q_1\, Q_2 \ldots Q_s\, \eta$. Felhasználva a $Q \Rightarrow u$ szabályt, kapjuk $\beta_i' = \gamma u\, \eta$.

11.1. Hatókörös összetett értékű funkcionális függőség

Az eddigi tárgyalásban a 2. típusú nemterminális szimbólumokat helyettesítő terminális szimbólumokon az adatbázis terminológiában értéknek nevezett adatokat értettünk. Ezek az értékek a szóban forgó attribútum (a helyettesített nemterminális szimbólum) által meghatározott értéktartományból

(az attribútum típusa) kerülnek kiválasztásra. Így egy 2. típusú nemterminális szimbólum annyi helyettesítési szabály bal oldalán szerepel, ahány terminális szimbólum helyettesítheti, illetve, a 11.3. Megjegyzés szerint, egyetlen olyan helyettesítési szabály bal oldalán áll, amelyben a jobb oldalt az illető nemterminális szimbólumot helyettesítő terminális szimbólumok diszjunkciója áll. A továbbiakban azt a megoldást választjuk (a 11.2. Példával demonstrálva), hogy a terminális szimbólumok a 2. típusú nemterminálisok értéktartományát nevezik meg, és a terminálisokból álló mondatokból kiértékeléssel (valuation) kapjuk az értéktartományok adataiból összefűzött mondatokat. A terminális szimbólumokhoz értékeket rendelünk (egy nem üres D halmazból választva) a következők szerint. Legyen $u \in T$ terminális szimbólum, akkor legyen $D(u) \in D$ egy halmaz úgy, hogy ha $u, v \in T, u \neq v$ akkor $D(u) \cap D(v) = \emptyset$.
A $val : u \in T \mapsto D(u)$ leképezés független kiválasztással rendel értéket a terminális szimbólumokhoz, vagyis ha a nemterminális szimbólumok egy sorozatához rendelünk értékeket akkor az egyes hozzárendelések egymástól függetlenül, autonóm módon történnek, azaz, $val(u_1) \neq val(u_2)$ bekövetkezhet akkor is, ha $u_1 = u_2$. Nyilván, ha $u_1 \neq u_2$ akkor $val(u_1) \neq val(u_2)$ (mert az értéktartományok diszjunktak).
Egy $\omega \in L(A)$ esetén, legyen $\omega = \{u_1 u_2 \dots u_n\}$ akkor $val(\omega) = \{v_1 v_2 \dots v_n\}$ az ω egy kiértékelése, ahol $v_i = val(u_i), 1 \leq i \leq n$.

11.2 Definíció (Összetett értékű sor)
Legyen $G=(N,T,S,P)$ egy $ECFGR$, legyen $A \in N$ nemterminális szimbólum és legyen $\alpha \in L_A, \alpha = \{B_1\ B_2\ \dots\ B_n\}$ nemterminális jelek sorozata és legyen $\omega \in L(\alpha), \omega = \{\omega_1\ \omega_2\ \dots\ \omega_n\}$ úgy, hogy mindegyik $\omega_i \in L(B_i)$ terminális jelek sorozata, ($1 \leq i \leq n$) akkor azt mondjuk, hogy α az A hatókörű (scope), összetett értékű duális mondat, a B_i-k az α összetett értékű attribútumai, ω_i a B_i-hez társított elemi összetett értékű sor és ω az A összetett értékű (complex valued) sora. Jelölje t_{CV} az ω egy összetett értékű kiértékelését, akkor $t_{CV} = val_{CV}(\omega)$.
Azt mondjuk, hogy $\alpha \in L_A$ az A hatókör egy összetett sortípusának $ECFGR$-jellegű sémája és L_A az A hatókör mint adatbázis összetett típusú sémája. Ha I az A összetett értékű sorai összetett értékű kiértékeléseinek (véges) halmaza, akkor azt mondjuk, hogy I az A előfordulása.

Ha $\omega = \{\omega_1\ \omega_2\ \dots\ \omega_n\}$ akkor $val(\omega) = \{val(\omega_1)\ val(\omega_2)\ \dots\ val(\omega_n)\}$.

Felidézzük a 11.1. Definícióból, hogy ha $\alpha = \{B_1\ B_2\ \dots\ B_n\}; B_j \in N; 1 \leq j \leq n$ akkor az $\alpha \Rightarrow \omega$ kifejezésen a $B_j \Rightarrow \omega_j; 1 \leq j \leq n$ helyettesítések egyidejű, egyszeri végrehajtását értjük ($\omega = \{\omega_1\ \omega_2\ \dots\ \omega_n\}$). $\alpha \Rightarrow^+ \omega$-t a természetes módon értelmeztük.

11.3 Definíció (Elemi összetett értékű sor)
Legyen $G=(N,T,S,P)$ *egy ECFGR, legyen* $B\in N$ *nemterminális szimbólum és legyen* $\omega \in L(B)$ *egy a* B*-hez társított elemi összetett sor, legyen* $\omega = =\{u_1\ u_2\ \ldots\ u_m\}; u_i\in T; 1\leq i\leq m$*. Van olyan* $\alpha\in L_B, \alpha=\{B_1\ B_2\ \ldots\ B_n\}; B_j\in \in N; 1\leq j\leq n$*, hogy* $\alpha \Rightarrow^+ \omega$*. Akkor minden* $1\leq i\leq m$*-re van* $1\leq j\leq n$ *úgy, hogy* $X_1^i = B_j, X_2^i, \ldots, X_{k_i}^i \Rightarrow u_i$ *(ahol az* $X_{k_r}^i$*-k 2. típusú, a többiek 1. típusú nemterminálisok), úgy hogy* $X_r^i \in \beta_r^j$ *valamely* $1\leq j\leq n$*-re, ahol* $B_j \Rightarrow^r \beta_r^j$*. A* $T_1 : D(u_1), T_2 : D(u_2), \ldots, T_m : D(u_m)$ *kifejezést az* ω *elemi összetett sor konstruktorának, a* $T_i = X_1^i.X_1^i \ldots X_{k_i}^i$ *kifejezést az* u_i *elemi attribútum típusának nevezzük.*

11.2. Példa. *A BetuSzam.dtd sémaleírás (29. ábra) kiterjesztett környezetfüggetlen nyelv grammatikájának felel meg, ha az összetett XML elemneveket 1. típusú nemterminális szimbólumoknak tekintjük, az egyszerű, szöveges értékű elemek nevei lesznek a 2. típusú nemterminális szimbólumok (minden elemnév nagybetűvel kezdődik). A 2. típusú nemterminálisok kisbetűvel írt változatai lesznek a terminális szimbólumok.*

11.4 Megjegyzés
A 2. típusú nemterminális szimbólumok és a terminális értékek (szimbólumok) közé iktatott tipizált nemterminális szimbólumokat azért vezetjük be (lényegében a 2. típusú nemterminális szimbólumok megkettőzésével), hogy az RFD modellben alkalmazott megoldás (szabályséma) helyett egy másik, a szabálysémával ekvivalens megoldást találjunk a nagyszámú, csak a terminális szimbólumban különböző helyettesítési szabály egyszerű megadására.

$G = (N, T, S, P)$*, ahol*
$N = \{Nevek, Név, Index, Id, Adat, Jel, Jel1, Betű, Szám\}$ *a nemterminális szimbólumok,*
$T = \{jel1, index, id, betű, szám\}$ *a terminális szimbólumok,*
$S = Nevek$ *a kezdő szimbólum,*
P=$\{Nevek \Rightarrow R_{Nevek}, Név \Rightarrow R_{Név},$
$Adat \Rightarrow R_{Adat}, Jel \Rightarrow R_{Jel}\}$
az első csoportba tartozó helyettesítési szabályok,
$Jel1 \Rightarrow jel1$
$Index \Rightarrow index$
$Id \Rightarrow id$
$Betű \Rightarrow betű$
$Szám \Rightarrow szám$
a második csoport helyettesítési szabályai.
Az első csoportba tartozó helyettesítési szabályokban hivatkozott reguláris kifejezések:

R_{Nevek}=<*Név*>$^+$
$R_{Név}$=(<*Index*> ∘ <*Id*> ∘ <*Adat*>)*
R_{Adat}=<*Jel*> ∘ <*Betű*> ∘ <*Szám*>
R_{Jel}=<*Jel1*> + <*Név*>

A terminális szimbólumokhoz tartozó értéktartományok:

$D\left(jel1\right) = \{NévX, NévY, \ldots, \}$
$D\left(index\right) = \{1,2,3, \ldots, \}$
$D\left(id\right) = \{10,20, \ldots, \}$
$D\left(betű\right) = \{A, B, \ldots, \}$
$D\left(szám\right) = \{2,3, \ldots, \}$

A 30. ábrán látható előfordulásban (XML dokumentum) a Név nemterminális szimbólumból generált három mondat látható: az 1,2 és 3 Index értékű elemek. Ezen elemek a Név nemterminális szimbólumhoz társított elemi összetett sorokat reprezentálják. Az elemi összetett sorok ($\omega_1, \omega_2, \omega_3$, az elemi összetett sorok egy-egy kiértékelése t_1, t_2, t_3) konstruktorai legyenek rendre K_1, K_2, K_3:

$K_1 = \langle T_1^1 : D\left(index\right), T_2^1 : D\left(id\right), T_3^1 : D\left(jel1\right), T_4^1 : D\left(betű\right), T_5^1 : D\left(szám\right)\rangle$
ahol
$T_1^1 = Index$
$T_2^1 = Id$
$T_3^1 = Adat.Jel.Jel1$
$T_4^1 = Adat.Betű$
$T_5^1 = Adat.Szám$
$\omega_1 = \lfloor\langle index \mid id \mid jel1 \mid betű \mid szám\rangle\rceil$
$t_1 = \lfloor\langle 1 \mid 10 \mid NévX \mid A \mid 2\rangle\rceil$
$K_2 = \langle T_1^2 : D\left(index\right), T_2^2 : D\left(id\right), T_3^2 : D\left(index\right),$
$T_4^2 : D\left(id\right), T_5^2 : D\left(jel1\right), T_6^2 : D\left(betű\right),$
$T_7^2 : D\left(szám\right), T_8^2 : D\left(betű\right), T_9^2 : D\left(szám\right)\rangle$
ahol
$T_1^2 = Index$
$T_2^2 = Id$
$T_3^2 = Adat.Jel.Név.Index$
$T_4^2 = Adat.Jel.Név.Id$
$T_5^2 = Adat.Jel.Név.Adat.Jel.Jel1$
$T_6^2 = Adat.Jel.Név.Adat.Betű$
$T_7^2 = Adat.Jel.Név.Adat.Szám$

$T_8^2 = Adat.Betű, T_9^2 = Adat.Szám$
$\omega_2 = \lfloor\langle index \mid id \mid index \mid id \mid jel1 \mid betű \mid szám \mid betű \mid szám\rangle\rceil$
$t_2 = \lfloor\langle 2 \mid 20 \mid 3 \mid 10 \mid NévY \mid B \mid 3 \mid B \mid 3\rangle\rceil$
$K_3 = \langle T_1^3 : D\,(index)\,, T_2^3 : D\,(id)\,, T_3^3 : D\,(jel1)\,, T_4^3 : D\,(betű)\,, T_5^3 : D\,(szám)\rangle$
ahol
$T_1^3 = Index$
$T_2^3 = Id$
$T_3^3 = Adat.Jel.Jel1$
$T_4^3 = Adat.Betű$
$T_5^3 = Adat.Szám$
$\omega_3 = \lfloor\langle index \mid id \mid jel1 \mid betű \mid szám\rangle\rceil$
$t_3 = \lfloor\langle 3 \mid 10 \mid NévY \mid B \mid 3\rangle\rceil$

Reguláris nyelvekre értelmeztük a funkcionális függőséget a 6. Fejezetben, a szintaktikus definíciót a nyelvet elfogadó automata gráfján adtuk meg, a függőség szemantikáját a nyelv mondatain ellenőriztük. Ezt a definíciót most kiterjesztjük az $ECFGR$-re úgy, hogy az FD-k szintaktikáját egy reguláris kifejezésen adjuk meg (csupán egy helyettesítési lépést alkalmazva), de a függőség szemantikájához az $ECFGR$ levezetési fa teljes mélységét felhasználjuk. Ezzel a szintaktikus megszorítással a gyakorlatban előforduló XML alkalmazások többsége kezelhető, "horizontálisan" összekapcsolt elemeket feltételezve, és a függőségek logikai implikációja kvadratikus időben eldönthető marad.
A duális nyelvet a 4.1. Definíció szerint kapjuk. A funkcionális függőség szintaktikus leírásához mindegyik duális mondaton két részsorozatot kell kiválasztanunk, egyet a bal, egyet a jobb oldal számára.

Legyen $G = (N, T, S, P)$ egy $ECFGR$, legyen $A \in N$ és legyen M_A az L_A nyelvet elfogadó automata. Legyen $\mathrm{w} \in L_A$ és legyen $\mathrm{w} = v_1 v_2 \ldots v_n$ akkor $walk\,(\mathrm{w}) = (v_1, v_2, \ldots, v_n)$ az M_A egy bejárása.

11.4 Definíció (Hatókörös kiválasztás)
Legyen $Y = (Y_1, Y_2)$ az A hatókörre vonatkozó kijelölés M_A felett, legyen $\mathrm{w} \in \in L_A$ és $walk\,(\mathrm{w}) = (v_1, v_2, \ldots, v_n)$ az M_A egy bejárása. Az Y_1 szimbólumait a bejárás sorrendjében választjuk ki (ha a bejárás érinti azokat). Minden $e \in Y_2$ él esetén, ha az él a végpontjai között a legrövidebb úton záródik, akkor ezt a két végpontot a bejárás sorrendjében kiválasztjuk (ha érintjük őket). Azaz, a záródó út két végpontja A és B ($A = v_i, B = v_j$ valamely $1 \leq i < j \leq n$-re) akkor azt az utat választjuk, amely sem A-t sem B-t nem tartalmazza. Y_2 csúcsait minden érintésnél kiválasztjuk a $walk\,(\mathrm{w})$ úton. A kiválasztás eredménye az (esetleg üres) $\mathrm{w}\,[Y] = v_{i_1} v_{i_2} \ldots v_{i_k}$ jelsorozat ($1 \leq i_1 < i_2 < \ldots < i_k \leq n\,(k \geq 0)$).

Legyen $\mathrm{w} \in L_A$ és legyen $\mathrm{w} = v_1 v_2 \ldots v_n$ akkor $walk\,(\mathrm{w}) = (v_1, v_2, \ldots, v_n)$ M_A bejárása, legyen $\omega \in L\,(\mathrm{w})$ és legyen $t = val\,(\omega)$ sor A-ban. A $\mathrm{w}\,[Y]$ jelsorozatot olyan "attribútumok" sorozataként értelmezzük amelyek a t sort a $t\,[Y] = val\,(\omega\,[Y])$ értékekre (értékek sorozatára) vetítik, azaz, legyen $\mathrm{w}\,[Y] = = v_{i_1} v_{i_2} \ldots v_{i_k}$ $(1 \leq i_1 < i_2 < \ldots < i_k \leq n\,(k \geq 0)$ és legyen $\omega = \lfloor \omega_1\, \omega_2\, \ldots\, \omega_n \rceil$ akkor $t\,[Y] = val\,(\omega_{i_1})\, val\,(\omega_{i_2}) \ldots val\,(\omega_{i_k})$.
Ha $\mathrm{w}\,[Y] = \epsilon$, akkor $t\,[Y] = \epsilon$ is teljesül.

Az L_A reguláris nyelven is definiálhatunk funkcionális függőséget az M_A gráf felhasználásával a 6. Fejezetben leírtak szerint, az A nemterminális szimbólum ezen funkcionális függőség szempontjából is a hatókör szerepét játssza.

11.5 Definíció (Hatókörös összetett értékű funkcionális függőség) *Legyen A hatókör a G $ECFGR$-re vonatkoztatva és legyen M_A az L_A-t elfogadó véges automata. Legyenek $X = (X_1, X_2)$ és $Y = (Y_1, Y_2)$ kijelölések M_A felett. Az M_A felett definiált (A-hatókörű) környezetfüggetlen funkcionális függőség (CFD, vagy FD_A) egy $X \to Y$ formájú kifejezés. Az A egy (véges) I adatbázis előfordulása kielégíti $X \to Y$-t (jelölve $I \models X \to Y$), ha bármely két $t_1, t_2 \in I$ sorra $t_1[X]$=$t_2[X]$ csak akkor teljesül, ha $t_1[Y]$=$t_2[Y]$ is igaz. Az $Y = M_A$ kulcs függőségnek nevezzük.*

11.3. Példa. *A 9.3 példában egy telefonhálózat előfizetőinek adatbázisán demonstráltuk a reguláris funkcionális függőséget abban az esetben, amikor az igénybe vett szolgáltatások leírására összetett adattípust használtunk. Bevezettük a Szolgáltatás összetett elemet:*

```
<!ELEMENT Előfizető (Név,T_év,(V_év,V_kód)*,Szolgáltatás,Díj)>
<!ELEMENT Szolgáltatás (Dsl,Iptv?,Telefon?)>
```

és megadtunk egy RFD-t arra az épségi megszorításra, hogy az Előfizető által havonta fizetendő Díj az adott Előfizető által igénybe vett szolgáltatások függvénye:

$$RFD: (\{Szolgáltatás\}, (\{\}, \{\})) \to (\{Díj\}, (\{\}, \{\}))$$

Megemlítettük, hogy a fenti séma csak korlátozottan felel meg a reguláris funkcionális függőség követelményeinek, mivel a Szolgáltatás összetett elem, a három gyermek elemnevei nem szerepelhetnek nemterminális szimbólumként (a $Szolgáltatás \Rightarrow Dsl\ Iptv^?\ Telefon^?$ helyettesítési szabály nem illik reguláris grammatikába), így viszont a terminális értékek típus nélkül maradhatnak

(mivel Iptv és Telefon opcionálisak), így például az („Alap”„Alap”) összetett érték tartozhat a (Dsl,Iptv) de a (Dsl,Telefon) elempárokhoz is, így két sorban a Szolgáltatás attribútum értékei látszólag egyenlőek lehetnek, de típussal ellátva már különböznének (pl. ha az egyikben csak Iptv, a másikban csak Telefon szerepel). Ezt a problémát a kiterjesztett környezetfüggetlen nyelvekre alapozott modell kezelni tudja.
Az Előfizetők adatbázis összetett elemtípusokat tartalmazó sémaleírását látjuk a 27. ábrán, a 28. ábra mutat egy, a sémának megfelelő előfordulást (XML dokumentumot) XML text formátumban, a 26. ábra az XML dokumentum szerkezetét mutatja, jól látható az összetett adatok beágyazódása a hierarchikus szerkezet szerint. Ez az XML dokumentum adattartalmát tekintve szinte teljesen azonos a 9.3. Példában megadottal (egy új adatelem - Vonal - szerepel, az is csak egyetlen Telefon adatelem részeként, de a séma jelentősen változott, két új szint (Kontroll és Szolgáltatás) jelent meg, a séma fő összetett típusú elemét, Előfizető-t, olyan reguláris kifejezés írja le, amely nem reguláris nyelvet, hanem kiterjesztett környezetfüggetlen nyelvet generál, a 11.2. Példa jelölésmódját alkalmazva:

$G = (N, T, S, P)$, *ahol*
$N = \{Előfizető, Név, T\_év, Kontroll, Szolgáltatás, V\_év, V\_kód,$
$Dsl, Iptv, Telefon, Vonal, Csomag, Díj\}$ *a nemterminális szimbólumok,*
$T = \{név, t\_év, v\_év, v\_kód, dsl, iptv, vonal, csomag, díj\}$ *a terminális szimbólumok,*
$S = Előfizető$ *a kezdő szimbólum,*
$P = \{Előfizető \Rightarrow R_{„Előfizető”}, Kontroll \Rightarrow R_{Kontroll},$
$Szolgáltatás \Rightarrow R_{Szolgáltatás}, Telefon \Rightarrow R_{Telefon}\}$ *az első csoportba tartozó helyettesítési szabályok,*
$Név \Rightarrow név$
$T\_év \Rightarrow t\_év$
$V\_év \Rightarrow v\_év$
$V\_kód \Rightarrow v\_kód$
$Dsl \Rightarrow dsl$
$Iptv \Rightarrow iptv$
$Vonal \Rightarrow vonal$
$Csomag \Rightarrow csomag$
$Díj \Rightarrow díj$ *a második csoport helyettesítési szabályai.*
Az első csoportba tartozó helyettesítési szabályokban hivatkozott reguláris kifejezések:
$R_{„Előfizető”} = <Név> \circ <T\_év>? \circ <Kontroll>^* \circ <Szolgáltatás> \circ <Díj>$
$R_{Kontroll} = <V\_év> \circ <V\_kód>$

$R_{Szolgáltatás}$=<*Dsl*> ∘ <*Iptv*>*?* ∘ <*Telefon*>*?*
$R_{Telefon}$=<*Vonal*> ∘ <*Csomag*>

Az ElőfizetőkC összetett elemtípusokat tartalmazó adatbázison is érvényesül az az épségi megszorítás, hogy az Előfizető által havonta fizetendő Díj az adott Előfizető által igénybe vett szolgáltatások függvénye. Ezt a megszorítást az ECFGR modellben az Előfizető hatókörre vonatkozó CFD-vel írhatjuk le. Ez a CFD szintaktikailag az Előfizető adatbázison értelmezett, fentebb megadott RFD-vel azonos:

$CFD1: X \to Y$, *ahol* $X = (\{Szolgáltatás\}, (\{\}, \{\}))$
és $Y = (\{Díj\}, (\{\}, \{\}))$
a $CFD1$ *jobb és bal oldalát leíró kijelölések* $M_{„Előfizető"}$ *felett.*

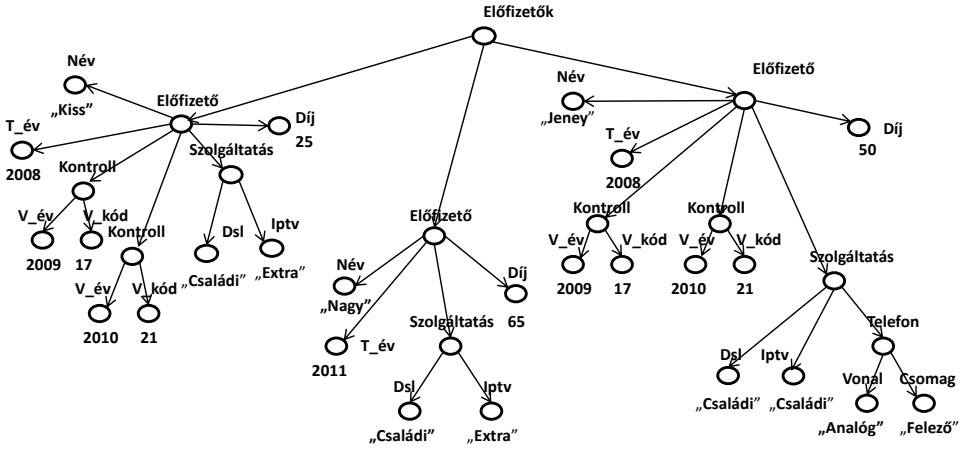
Legyen most az Előfizető hatókör előfordulása a 28. ábra XML dokumentumában adott három elem. Alkalmazva a 11.2. Példához hasonlóan felépíthető összetett sor konstruktorokat, három összetett értékű sort és ezek egy-egy kiértékelését kapjuk:

ω_1=	**Név**	**T_év**	**V_év**	**V_kód**	**V_év**	**V_kód**	**Dsl**	**Iptv**	**Díj**
t_1=	Kiss	2008	2009	17	2010	21	Családi	Extra	25
ω_2=	**Név**	**T_év**	**Dsl**	**Iptv**	**Díj**				
t_2=	Nagy	2011	Családi	Extra	65				
ω_3=	**Név**	**T_év**	**V_év**	**V_kód**	**V_év**	**V_kód**	**Dsl**	**Iptv**	
	Vonal	**Csomag**	**Díj**						
t_3=	Jeney	2008	2009	17	2010	21	Családi	Családi	
	Analóg	Fizető	25						

4. táblázat. Az *ElőfizetőkC* adatbázis *Előfizető* hatókörének előfordulásai

$\omega_1[X] = \lfloor dsl \mid iptv \rceil, \omega_1[Y] = díj$
$t_1[X] = \lfloor Családi \mid Extra \rceil, t_1[Y] = 25$
$\omega_2[X] = \lfloor dsl \mid iptv \rceil, \omega_2[Y] = díj$
$t_2[X] = \lfloor Családi \mid Extra \rceil, t_2[Y] = 65$
$\omega_3[X] = \lfloor dsl \mid iptv \mid vonal \mid csomag \rceil, \omega_3[Y] = díj$
$t_3[X] = \lfloor Családi \mid Családi \mid Analóg \mid Fizető \rceil, t_3[Y] = 25$

Az előfordulás nem elégíti ki $CFD1$*-et, mert* $t_1[X] = \lfloor Családi \mid Extra \rceil =$
$= t_2[X]$ *de* $t_1[Y] = 25 \neq 65 = t_2[Y]$ *ti ki* $CFD1$*-et, mert* $t_1[X] = \lfloor Családi \mid Extra \rceil =$
$= t_2[X]$ *de* $t_1[Y] = 25 \neq 65 = t_2[Y]$



26. ábra. XML példa dokumentum telekom előfizetői adatokra (ECFGR modell)

11.6 Definíció (Hatókörös összetett értékű FD implikációja) *Legyen A hatókör a G $ECFGR$-ben és legyen M_A a társított véges automata. Legyen Σ a FD_A-k halmaza és legyen σ egy FD_A (valamennyien M_A felett), akkor Σ implikálja σ-t (jelölve $\Sigma \models \sigma$) ha az A minden I (véges) adatbázis előfordulására amelyek kielégítik Σ-t, $I \models \sigma$ is teljesül.*

11.5 Megjegyzés (Hatókörös összetett értékű FD implikációja) *A reguláris funkcionális függőséget megalapozó reguláris nyelvet egy véges automata fogadja el, az $ECFGR$ modellel leírt nyelvet általános esetben verem automata ismeri fel. A hatókörös összetett értékű funkcionális függőségek logikai implikációjának eldöntéséhez mégis felhasználhatjuk a 6.1. Algoritmust, mert az A hatókörhöz tartozó L_A reguláris nyelvet duális nyelvnek tekintve, a nyelv szavait alkotó nemterminálisokat összetett értékükkel helyettesítve, ezen összetett értékek reguláris nyelvét kapjuk. Az L_A duális nyelven adjuk meg a funkcionális függőség szintaktikus leírását, az összetett értékek reguláris nyelvén vizsgálhatjuk a funkcionális függőség teljesülését (szemantika). A reguláris modelltől eltérően az $ECFGR$ modellben gondot okozhat, ha az A hatókört meghatározó R_A reguláris kifejezés valamelyik nemterminális elemének levezetésében az A hatókör előfordul. Miután csak véges jelsorozatokat generálunk, ezért ez az eset kezelhető, a levezetés során bejövő újabb hatókörök egyrészt beépülnek a levezetésben megelőzően keletkező A hatókörök össze-*

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
               elementFormDefault="unqualified">
  <xs:element name="ElőfizetőkC">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded" minOccurs="1">
        <xs:element name="Előfizető" type="t_Előfizető"
                  minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="t_Kontroll">
    <xs:sequence>
      <xs:element name="V_év" type="xs:integer"/>
      <xs:element name="V_kód" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="t_Telefon">
    <xs:sequence>
      <xs:element name="Vonal" type="xs:NCName"/>
      <xs:element name="Csomag" type="xs:NCName"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="t_Szolgáltatás">
    <xs:sequence>
      <xs:element name="Dsl" type="xs:NCName"/>
      <xs:element minOccurs="0" maxOccurs="1" name="Iptv"
                  type="xs:NCName"/>
      <xs:element minOccurs="0" maxOccurs="1" name="Telefon"
                  type="t_Telefon"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="t_Előfizető">
    <xs:sequence>
      <xs:element name="Név" type="xs:NCName"/>
      <xs:element name="T_év" type="xs:integer"/>
  <xs:element name="Kontroll"
              minOccurs="0" maxOccurs="unbounded"
              type="t_Kontroll"/>
      <xs:element name="Szolgáltatás" type="t_Szolgáltatás"/>
      <xs:element name="Díj" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

27. ábra. Az *ElőfizetőkC* összetett típusokra épülő sémaleírása (t_com_REGA.xsd)

```
<?xml version="1.0"?>
<ElőfizetőkC xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    xsi:noNamespaceSchemaLocation="t_com_REGA.xsd">
<Előfizető>
<Név>Kiss</Név>
<T_év>2008</T_év>
<Kontroll><V_év>2009</V_év><V_kód>17</V_kód></Kontroll>
<Kontroll><V_év>2010</V_év><V_kód>21</V_kód></Kontroll>
<Szolgáltatás><Dsl>Családi</Dsl><Iptv>Extra</Iptv></Szolgáltatás>
<Díj>25</Díj>
</Előfizető>
<Előfizető>
<Név>Nagy</Név>
<T_év>2011</T_év>
<Szolgáltatás><Dsl>Családi</Dsl><Iptv>Extra</Iptv></Szolgáltatás>
<Díj>65</Díj>
</Előfizető>
<Előfizető>
<Név>Jeney</Név>
<T_év>2008</T_év>
<Kontroll><V_év>2009</V_év><V_kód>17</V_kód></Kontroll>
<Kontroll><V_év>2010</V_év><V_kód>21</V_kód></Kontroll>
<Szolgáltatás><Dsl>Családi</Dsl><Iptv>Családi</Iptv>
<Telefon><<Vonal>Analóg</Vonal><Csomag>Felező</Csomag>
</Telefon>
</Szolgáltatás>
<Díj>25</Díj>
</Előfizető>
</Előfizetők>
```

28. ábra. XML példa dokumentum telekom előfizetői adatokra (ECFGR modell)

tett értékeibe, másrészt az A hatókör autonóm előfordulásaiként is figyelembe vesszük őket, az alábbi példa szerint.

11.4. Példa. *A BetuSzam.dtd sémaleírás (29. ábra) Név elemének deklarációja a* $Név \longmapsto Adat \longmapsto Jel \longmapsto Név$ *kört hozza létre a DTD gráfjában. Ez a kör a 30. ábrán látható előfordulásban (XML dokumentum) a Név három előfordulását alapozza meg: az 1,2 és 3 Index értékű elemekét. Ezen elemek adataiból a Név elem mint hatókör három előfordulása, azaz három sor épül fel a 11.2. Példa jelöléseit felhasználva (5. táblázat).*

< Legyen CFD2 hatókörös összetett értékű funkcionális függőség a Név

ω_1=	**index**	**id**	**jel1**	**betű**	**szám**			
t_1=	1	10	NévX	A	2			
ω_2=	**index**	**id**	**index**	**id**	**jel1**	**betű**	**szám**	**betű**	**szám**
t_2=	2	20	3	10	NévY	B	3	B	2
ω_3=	**index**	**id**	**jel1**	**betű**	**szám**				
t_3=	3	10	NévY	B	3				

5. táblázat. A *Nevek* adatbázis *Név* hatókörének előfordulásai

elem mint hatókör felett.

$CFD2$*: $X \to Y$, ahol* $X = (\{Id\}, (\{\}, \{\}))$
és $Y = (\{Adat\}, (\{\}, \{\}))$
a $CFD2$ *jobb és bal oldalát leíró kijelölések* $M_{Név}$ *felett.*
$\omega_1[X] = Id, \omega_1[Y] = \lfloor jel1 \mid betű \mid szám \rceil$
$t_1[X] = 10, t_1[Y] = \lfloor NévX \mid A \mid 2 \rceil$
$\omega_2[X] = id, \omega_2[Y] = \lfloor index \mid id \mid jel1 \mid betű \mid szám \mid betű \mid szám \rceil$
$t_2[X] = 20, t_2[Y] = \lfloor 3 \mid 10 \mid NévY \mid B \mid 3 \mid B \mid 2 \rceil$
$\omega_3[X] = id, \omega_3[Y] = \lfloor betű \mid szám \rceil$
$t_3[X] = 10, t_3[Y] = \lfloor NévY \mid B \mid 3 \rceil$

Az előfordulás nem elégíti ki $CFD2$*, mert* $t_1[X] = 10 = t_3[X]$ *és* $\omega_1[Y] = = \lfloor betű \mid szám \rceil = \omega_3[Y]$*, de* $t_1[Y] = \lfloor NévX \mid A \mid 2 \rceil \neq \lfloor NévY \mid B \mid 3 \rceil = t_3[Y]$*. Definiáljunk az Adat hatókörön funkcionális függőséget:*

$CFD3$*: $X \to Y$, ahol* $X = (\{Betű\}, (\{\}, \{\}))$
és $Y = (\{Szám\}, (\{\}, \{\}))$
a $CFD3$ *jobb és bal oldalát leíró kijelölések az Adat hatókör felett.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Nevek (Név)+>
<!ELEMENT Név (Index,Id,Adat)*>
<!ELEMENT Adat (Jel,Betű,Szám)>
<!ELEMENT Jel (Jel1 | Név)*>
<!ELEMENT Jel1 (#PCDATA)>
<!ELEMENT Index (#PCDATA)>
<!ELEMENT Id (#PCDATA)>
<!ELEMENT Betű (#PCDATA)>
<!ELEMENT Szám (#PCDATA)>
```

29. ábra. A *Nevek* rekurzív elemeket tartalmazó sémaleírás (BetuSzam.dtd)

Az Adat hatókörnek is három előfordulása és ezek egy-egy kiértékelése adott $(\omega_4, \omega_5, \omega_6; t_4, t_5, t_6)$ $\omega_4[X] = \omega_5[X] = \omega_6[X] = betű$
$\omega_4[Y] = \omega_5[Y] = \omega_6[Y] = szám$
$t_4[X] = A, t_4[Y] = 2$
$t_5[X] = B, t_5[Y] = 2$
$t_6[X] = B, t_6[Y] = 3$

Az előfordulás nem elégíti ki $CFD3$*-at, mert* $B{=}t_5[X]{=}t_6[X]$*, de* $2{=}t_5[Y]{\neq}{=}t_6[Y] = 3$.

11.1. Algoritmus. *Algoritmus hatókörös összetett értékű funkcionális függőségek implikációjának eldöntésére.*
Input: az A hatókör L_A *reguláris nyelvét eldöntő* $M_A = (V, E)$ *gráf, az* M_A *feletti* $\sigma : X \to Y$ *(ahol* $X{=}(X_1, X_2)$ *és* $Y{=}(Y_1, Y_2)$*) funkcionális függőség és ezek* Σ *halmaza*
Output: igaz, ha $\Sigma \models \sigma$*, hamis egyébként*

Az algoritmus a 6.1. Algoritmushoz hasonlóan működik, mert a funkcionális függőségekben csak a hatókör közvetlen beépülő nemterminálisai vesznek részt, ezek (összetett) értékei egymástól függetlenek (11.5. Megjegyzés).

11.1 Propozíció (Hatókörös CFD implikációja)
Legyen A hatókör a G ECFGR-ben és legyen M_A *a hatókörhöz tartozó véges automata és legyen* $X \to Y$ *funkcionális függőség és* Σ *funkcionális függőségek halmaza* M_A *felett, úgy* $\Sigma{\models}X \to Y$ *akkor és csak akkor, ha a 11.1. Algoritmus az* M_A*,* Σ *és* $X \to Y$ *inputtal igaz eredményre jut.*

```
<?xml version="1.0"?><!DOCTYPE Nevek SYSTEM "BetuSzam.dtd">
<Nevek>
<Név>
<Index>1</Index>
<Id>10</Id>
<Adat><Jel><Jel1>NévX</Jel1></Jel><Betű>A</Betű><Szám>2</Szám></Adat>
</Név>
<Név>
<Index>2</Index>
<Id>20</Id>
<Adat>
<Jel>
<Név>
<Index>3</Index>
<Id>10</Id>
<Adat><Jel><Jel1>NévY</Jel1></Jel><Betű>B</Betű><Szám>3</Szám></Adat>
</Név>
</Jel>
<Betű>B</Betű><Szám>2</Szám>
</Adat>
</Név>
</Nevek>
```

30. ábra. XML példa dokumentum rekurzív elemekkel (ECFGR modell)

Bizonyítás
A bizonyítás gondolatmenete megegyezik 6.1. bizonyítással, a 11.5. Megjegyzéssel kiegészítve.

11.6 Megjegyzés (Hatókörös CFD implikációjának bonyolultsága)
A 11.1. Algoritmus kvadratikus időt használ fel a Σ-ban és $\sigma : X \to Y$-ban előforduló nemterminálisok száma szerint mérve.

11.2. Hatókörös egyszerű értékű funkcionális függőség

A 11.4 és 11.5 definíciók egy $ECFGR$ adott hatókörének közvetlen, nemterminális szimbólumokkal reprezentált komponensein értelmeztek funkcionális függőséget. A hatókör az $ECFGR$ egy helyettesítési szabályának bal oldalán álló nemterminális szimbólum, a szabály jobb oldalán álló reguláris kifejezés által generált nyelvet elfogadó véges automata állapotai megfelelnek a hatókör komponenseinek, ezért a funkcionális függőséget az automata állapotain specifikáljuk. Ezt a relatíve egyszerű esetet a reguláris funkcionális függőséghez hasonlóan kezelhetjük a levezetés során keletkező összetett értékek figyelembe vételével, így a funkcionális függőségek implikációjának eldöntésére is nyerünk hatékony algoritmust. Ha a funkcionális függőség specifikálásában a közvetlen beépülő elemeken túl leszármazott elemeket is megengedünk, bonyolultabb, az XFD-k definíciójához hasonló, fa-szerkezetű modellhez jutunk.

11.7 Definíció (Egyszerű értékek sora)
Legyen $G = (N, T, S, P)$ egy $ECFGR$, legyen $A \in N$ és legyen $\omega \in L(A), \omega = \lfloor \omega_1 \omega_2 \dots \omega_n \rceil, \omega_i \in T$ terminális jelek sorozata, azt mondjuk, hogy ω egyszerű értékű duális mondat az A hatókörre nézve, az ω_i-k az ω egyszerű értékű attribútumai, és az ω valamely t kiértékelése az A egy egyszerű értékű sora, jelölve $t = val(\omega)$.
A $T_A = \{\alpha | \alpha \in T; \exists \beta \in L(A); \alpha \in [\beta =]\}$ halmazt az A attribútumhalmazának nevezzük.
Azt mondjuk, hogy $L(A)$ a G-hez tartozó $ECFGR$-stílusú (egyszerű értékű) séma, és ha I az A (egyszerű értékű) sorainak véges halmaza, akkor I az A (egyszerű értékű) előfordulása.

Ha $\omega = \lfloor \omega_1 \omega_2 \dots \omega_n \rceil$ akkor $val(\omega) = \lfloor val(\omega_1)\, val(\omega_2) \dots val(\omega_n) \rceil$

11.7 Megjegyzés
Visszautalva a 11.4. Megjegyzésre, a 2. típusú nemterminális szimbólumok és a terminális értékek (szimbólumok) közé iktatott tipizált nemterminális szimbólumokat azért vezettük be, hogy az RFD modellben alkalmazott megoldás (szabályséma) helyett egy másik, a szabálysémával ekvivalens megoldást találjunk a nagyszámú, csak a terminális szimbólumban különböző helyettesítési

szabály egyszerű megadására. A hatókörös egyszerű értékű sorokat alkalmazó modellnél kézenfekvő, hogy ezek a tipizált „terminális" szimbólumok vegyék át az attribútumok szerepét.
Megtehetnénk, hogy az attribútumokat az RFD modellhez hasonlóan értelmezett duális mondatból vezetjük le: legyen $\omega \in L(A), \omega = \lfloor \omega_1 \omega_2 \ldots \omega_n \rceil, \omega_i \in T$ *terminális jelek sorozata, azt mondjuk, hogy* ω *egyszerű értékű sor az A hatókörre nézve. A 11.3. Megjegyzés szerint* ω *szimbólumai egy* $\mathrm{w} = \lfloor v_1 v_2 \ldots v_n \rceil ; v_i \in N_2$ *mondat szimbólumaiból helyettesítéssel keletkeztek, mégpedig úgy, hogy ezek a helyettesítések az* ω*-t generáló helyettesítési lánc végén, összefüggő sorozatban hajtódtak végre (bár sorrendjük közömbös). Ezt a* $\mathrm{w} \in N_2^*$ *mondatot az* ω *társított duális mondatának nevezzük, a* v_i *szimbólumok az* ω *egyszerű értékű attribútumai, az* $L(A)$ *mondataihoz társított duális mondatok összességét* $D(A)$*-val jelöljük.*
Az $N_A = \{\alpha | \alpha \in N_2; \exists \beta \in D(A); \alpha \in [\beta]\}$ *halmazt az A attribútumhalmazának nevezzük.*
Ez a megoldás azonban félrevezető: a $D(A)$ *nyelv csak akkor reguláris, ha az* $L(A)$ *nyelv is az, ez pedig, mivel G egy ECFGR, általában nem teljesül.*

Legyen $A \in N$, legyen $\omega \in L(A)$, legyen $Y \subseteq T_A$ és legyen $t = val(\omega)$ egy egyszerű értékű sor A-ban. Az $\omega[Y]$ jelsorozatot "attribútumok" sorozataként értelmezzük, amelyek a t sort a $t[Y] = val(\omega[Y])$ értékekre vetítik, azaz, legyen $\omega = \lfloor \omega_1 \omega_2 \ldots \omega_n \rceil$, legyen $\omega[Y] = \lfloor \omega_{i_1} \omega_{i_2} \ldots \omega_{i_k} \rceil$ $(1 \leq i_1 < i_2 < \ldots < i_k \leq n \; (k \geq 0)$ akkor és csak akkor, ha $\omega_{i_k} \in Y$, akkor $t[Y] = val(\omega_{i_1}) \, val(\omega_{i_2}) \ldots val(\omega_{i_k})$. Ha $\omega[Y] = \epsilon$, akkor $t[Y] = \epsilon$ is teljesül.

11.8 Definíció (Hatókörös egyszerű értékű funkcionális függőség) *Legyen A a G ECFGR hatóköre. Legyenek* $X \subseteq T_A$ *és* $Y \subseteq T_A$ *az A egyszerű értékű attribútumainak halmazai. Az A felett értelmezett egyszerű értékű funkcionális függőségnek* (SFD_A) *nevezzük az* $X \to Y$ *kifejezést. Az A egy (véges) I adatbázis előfordulása kielégíti az* $X \to Y$ *funkcionális függőséget (jelölve* $I \models X \to Y$*), ha bármely két* $t_1, t_2 \in I$ *sorra* $t_1[X] = t_2[X]$ *csak akkor teljesülhet, ha* $t_1[Y] = t_2[Y]$ *is igaz.*

11.5. Példa. *A 11.5. Példa BetuSzam adatbázisának sémaleírása (29. ábra) kiterjesztett környezetfüggetlen nyelv grammatikájának felel meg. A példa jelöléseit használva az Adat hatókör három egyszerű értékű sorát és ezek egy-egy egyszerű értékű kiértékelését olvashatjuk ki a 30. ábra XML dokumentumából:*

$\omega_4 = \lfloor \langle jel1 \mid betű \mid szám \rangle \rceil$
$t_4 = \lfloor \langle NévX \mid A \mid 2 \rangle \rceil$
$\omega_5 = \lfloor \langle jel1 \mid betű \mid szám \mid betű \mid szám \rangle \rceil$

$t_5 = \lfloor\langle NévY \mid B \mid 3 \mid B \mid 2\rangle\rceil$
$\omega_6 = \lfloor\langle jel1 \mid betű \mid szám\rangle\rceil$
$t_3 = \lfloor\langle NévY \mid B \mid 3\rangle\rceil$
Legyen SFD hatókörös egyszerű értékű funkcionális függőség az Adat elem mint hatókör felett.
SFD*: betű* $\rightarrow$ *szám*
Látható, hogy SFD ugyanazt a megszorítást írja le, mint a 11.4. Példa $CFD2$ függősége. Mint láthattuk, a példa XML dokumentumában adott előfordulás nem elégíti ki $CFD2t$, de kielégíti SFD-t, hiszen a funkcionális függőség bal oldala minden soron különböző:
$\omega_4[X] = \omega_6[X] =$ *betű*
$\omega_5[X] =$ *betű betű*
$\omega_4[Y] = \omega_6[Y] =$ *szám*
$\omega_5[Y] =$ *szám szám*
$t_4[X] = A, t_4[Y] = 2$
$t_5[X] = B\ B, t_5[Y] = 3\ 2$
$t_6[X] = B, t_6[Y] = 3$

11.8 Megjegyzés (Hatókörös SFD implikációjának bonyolultsága) *A hatókörös egyszerű értékű funkcionális függőség implikációja a relációs modellhez hasonlóan kezelhető, ha az A hatókör által generált $L(A)$ nyelv valamely mondatát egy reláció sémájának tekintjük. Az A hatókört magában foglaló G $ECFG$ átmenetdiagrammjában ($G = (N, T, S, P)$; S a gyökér, N elemei a belső csúcsok, T elemei a levelek) az A csúcshoz tartozó részfa felett valamelyik XFD modellt értelmezve, az SFD logikai implikációja az általános esetben is (a teljes $L(A)$ nyelvre kiterjesztve) kezelhető.*

Irodalomjegyzék

[1] Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)

[2] Abiteboul, S., Buneman, P., Suciu, D.: Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann Publishers (2000)

[3] Autebert, J.-M., Beauquier, J. Boasson, L.: Langages sur des alphabets infinis. Discrete Applied Mathematics 2, p. 1–20 (1980)

[4] Albert, J., Giammarresi, D.,Wood, D.: Normal form algorithms for extended context-free grammars. Theoretical Computer Science, Volume 267(1–2), p. 35–47 (2001)

[5] Amano, S., Libkin, L., Murlak, F.: XML schema mappings. In Proc. PODS, p. 33–42 (2009)

[6] Arenas, M., Libkin, L.: A normal form for XML documents. ACM Transactions on Database Systems, Volume 29(1), p. 195–232 (2004)

[7] Atzeni, P., Morfuni, N.: Functional dependencies and constraints on null values in database relations. Information and Control, Volume 70(1), p. 1–31 (1986)

[8] Berry, G., Sethi, R.: From regular expressions to deterministic automata. Theoretical Computer Science, Volume 48(3), p. 117–126 (1986)

[9] Bouyer, P. , Petit, A., Thérien, D.: An algebraic approach to data languages and timed languages. Information and Computation, Volume 182(2), p. 137–162 (2003)

[10] Brzozowski, J.A.: Derivatives of regular expressions, Journal of the ACM, Volume 11(4), p. 481–494 (1964)

[11] Buneman, P., Davidson, S. B., Fan, W., Hara, C. S., Tan, W. C.: Keys for XML. Computer Networks, Volume 39(5), p. 473–487 (2002)

[12] Buneman, P., Davidson, S. B., Fan, W., Hara, C. S., Tan, W. C.: Reasoning about keys for XML. Information Systems, Volume 28(8), p. 1037–1063 (2003)

[13] Champarnaud, J.-M., Ziadi, D.: Canonical derivatives, partial derivatives and finite automaton constructions. Theoretical Computer Science, Volume 289(1), p. 137–163 (2002)

[14] Chen, Y., Davidson, S. B., Zheng, Y.: Constraint Preserving XML Storage in Relations. In Proc. WEBDB, p. 7–12 (2002)

[15] Dassow, Jürgen, Vaszil, György: P finite automata and regular languages over countably infinite alphabets. In Proc. 7th international conference on Membrane Computing, p. 367–381 (2006)

[16] Davidson, S., Fan, W., Hara, C.: Propagating XML constraints to relations. Journal of Computer and System Sciences, Volume 73(3), p. 316–361 (2007)

[17] Fagin, R.: Functional dependencies in a relational database and propositional logic. IBM Journal of Research and Development, Volume 21(6), p. 543–544 (1977)

[18] Fan, W., M., Libkin: On XML integrity constraints in the presence of DTDs. Journal of the ACM, Volume 49(3), p. 368–406 (2002)

[19] Fan, W., Simeon, J.: Integrity constraints for XML. In Proc. PODS, p. 23–34 (2000)

[20] Fischer, P.C., Saxton, L.V., Thomas, S.J., Van Gucht, D.: Interactions between Dependencies and Nested Relational Structures, Journal of Computer and System Sciences, Volume 31(3), p. 343–354 (1985)

[21] Francez, N., Kaminski, M. An algebraic characterization of deterministic regular languages over infinite alphabets. Theoretical Computer Science, Volume 306(1–3), p.155-175 (2003)

[22] Glushkov, V.M.: The abstract theory of automata. Russian Mathematical Surveys, Volume 16, p. 1–53 (1961)

[23] Grumberg, O., Kupferman, O., Sheinvald, S.: Variable automata over infinite alphabets. In Proc. 4th international conference on Language and Automata Theory and Applications, p. 561–572 (2010)

[24] Hara, C. S., Davidson, S. B.: Reasoning about Nested Functional Dependencies. In Proc. PODS, p. 91–100 (1999)

[25] Hartmann, S., Link, S: More Functional Dependencies for XML. In Proc. ADBIS, p. 355–369 (2003)

[26] Hartmann, S., Link, S., Schewe, K-D.: Functional Dependencies over XML Documents with DTDs. Acta Cybernetica, Volume 17(1), (2005)

[27] Hartmann, S., Link, S., Schewe, K-D.: Axiomatisation of functional dependencies in the presence of records, lists, sets and multisets. Theoretical Computer Science, Volume 355(2), p. 167–196 (2006)

[28] Hartmann, S., Link, S: Characterising nested database dependencies by fragments of propositional logic. Annals of Pure and Applied Logic, 152(1–3), p. 84–106 (2008)

[29] Hartmann, S., Link, S.: Efficient reasoning about a robust XML key fragment. ACM Transactions on Database Systems, Volume 34(2), p. 1–33 (2009)

[30] Hartmann, S., Link, S: Numerical constraints on XML data. Information and Computation, Volume 208(5), p. 521–544 (2010)

[31] Hartmann, S., Köhler, H., Trinh, T.: On the existence of Armstrong data trees for XML functional dependencies. In Proc. FoIKS, p. 94–113 (2010)

[32] Hartmann, S., Link, S., Trinh, T.: Solving the Implication Problem for XML Functional Dependencies with Properties. Logic, Language, Information and Computation, In Proc. WoLLIC, p. 161–175 (2010)

[33] Kaminski, M., Francez, N.: Finite-memory automata. Theoretical Computer Science, Volume 134(2), p. 329–363 (1994)

[34] Kaminski, M., Tan, T.: Regular expressions for languages over infinite alphabets. Fundamenta Informaticae, Volume 69(3), p. 301-318 (2006)

[35] Kot, L., White, W.: Characterization of the Interaction of XML Functional Dependencies with DTDs. In Proc. ICDT, p. 119–133 (2007)

[36] Lee, D., Mani, M., Murata, M.: Reasoning about XML Schema Languages using Formal Language Theory. Technical Report, IBM Almaden Research Center, http://www.cs.ucla.edu/dongwon/paper. (2000)

[37] Libkin, L., Vrgoč, D.: Regular expressions for data words. In Proc. 18th international conference on Logic for Programming, Artificial Intelligence, and Reasoning, p. 274–288 (2012)

[38] Liu, C., Vincent, M. W., Liu, J.: Constraint Preserving Transformation from Relational Schema to XML Schema. World Wide Web, Volume 9(1), p. 93–110 (2006)

[39] Lv, T., Yan, P.: Mapping Relational Schemas to XML DTDs with Constraints. In Proc. IMSCCS, p. 528–533 (2006)

[40] Martens, W., Neven, F., Schwentick, T., Bex, G. J.: Expressiveness and complexity of XML Schema, ACM Transactions on Database Systems, Volume 31(3), p. 770–813, (2006)

[41] Molecular Modeling Database, National Center for Biotechnology Information http://www.ncbi.nlm.nih.gov/dtd/

[42] Murata, M., Lee, D., Mani, M., Kawaguchi, K.: Taxonomy of XML schema languages using formal language theory. ACM Transactions on Internet Technology, Volume 5(4), p. 660–704 (2005)

[43] Neven, F., Schwentick, Th., Vianu, V.: Finite state machines for strings over infinite alphabets. ACM TOCL Volume 5(3), p. 403-435 (2004)

[44] Nicaud, C., Pivoteau, C., Razet, B.: Average Analysis of Glushkov Automata under a BST-Like Model. In Proc. FSTTCS, p. 388-399 (2010)

[45] Otto, F.: Classes of regular and context-free languages over countably infinite alphabets. Discrete Applied Mathematics 12, p. 41–56 (1985)

[46] Sali, A., Schewe, K-D.: Counter-Free Keys and Functional Dependencies in Higher-Order Datamodels. Fundamenta Informaticae, Volume 70(3), p. 277–301 (2006)

[47] Sperberg-McQueen, C. M., Thompson, H.: XML Schema. Technical report, World Wide Web Consortium, http://www.w3.org/XML/Schema. (2005)

[48] Szabó, G. I., Benczúr, A.: Functional Dependencies on Extended Relations Defined by Regular Languages. In Proc. FoIKS, p. 385–404 (2012)

[49] Szabó, G.I., Benczúr, A.: Functional Dependencies on Symbol Strings Generated by Extended Context Free Languages In Proc. ADBIS,AISC 186, p. 253–264 (2012)

[50] Szabó, G.I., Benczúr, A.: Encapsulated Functional Dependencies for XML Design, Research Conference on Information Technology, 24–25 October, 2011, Pécs, Hungary, C129, ISBN 978-963-7298-46-2

[51] Szabó, G.I.: How Much is XML Involved in DB Publishing? Conference of PhD Students in Computer Science, June 29–July 2, 2010, Szeged Hungary

[52] Vincent, M. W., Liu, J.: Multivalued dependencies and a 4NF for XML. In Proc. CAISE, p. 14–29 (2003)

[53] Vincent, M.W., Liu, J., Liu, C.: Strong functional dependencies and their application to normal forms in XML. ACM Transactions on Database Systems, Volume 29(3), p. 445–462 (2004)

[54] Vincent, M.W., Liu, J., Mohania, M.K.: On the equivalence between FDs in XML and FDs in relations. Acta Informatica, Volume 44(3–4), p. 207–247 (2007)

[55] Wang, J.: A comparative study of functional dependencies for XML. In Proc. APWeb, p. 308–319 (2005)

[56] Wang, J., Topor, R. W.: Removing XML Data Redundancies Using Functional and Equality-Generating Dependencies. In Proc. ADC, p. 65–74 (2005)

[57] Watson, B. W.: A taxonomy of finite automata construction algorithms. Computing Science Note 93/43, Eindhoven University of Technology, The Netherlands (1994).

[58] Yu, C., Jagadish, H. V.: XML schema refinement through redundancy detection and normalization. The VLDB Journal, Volume 17(2), p. 203–223 (2008)

[59] Zhou, R., Liu, C., Li, J.: Holistic constraint-preserving transformation from relational schema into XML schema. In Proc. DSFAA, p. 4–18 (2008)

Összefoglalás

Ebben a dolgozatban az XML adatbázisokon értelmezett függőségekkel (épségi megszorításokkal) foglalkoztam egy új aspektusból. A dolgozat központi gondolata az XML fa-szerkezetű („kétdimenziós") modelljének egydimenziós, reguláris kifejezéssel leírható modellre való szűkítése, függőségeknek ezen a modellen, reguláris nyelv mondataiból felépített adatbázison való vizsgálata. A relációs adatbázis függőségeit az adatbázis sémáján elsőrendű logikai kifejezésekkel adhatjuk meg. A reguláris nyelveket kiterjesztett relációként értelmezve, általánosítottam a relációs adatbázist a kiterjesztett relációk adatbázisára, definiáltam a funkcionális függőséget ezen az adatbázison, a reguláris nyelvet elfogadó véges automata állapotait attribútumokként értelmezve (4.4.,4.5.,4.6. Definíciók). Az automata gráfján az elfogadó bejárások éleinek „színei" a reguláris nyelv, az érintett csúcsok a duális nyelv mondatait hozzák létre. A reguláris funkcionális függőség lényegében az automata két részgráfja közötti kapcsolat (6.3. Definíció). Az automata gráfjának színezésén alapuló (Chase-jellegű) 6.1. Algoritmust adtam a funkcionális függőségek logikai implikációjának kvadratikus időben való eldöntésére. Két gráfból - az automata gráfja és a gráf tranzitív lezártja - ellenpéldát felépítve (a függőség baloldalát képviselő részgráfok egyszinűek, a jobboldaléi eltérőek a két gráfon) az algoritmus az implikáló függőséghalmaz elemeit alkalmazva próbálja „rendbe hozni" az ellenpéldát. A dolgozat fő eredménye a 6.1. Algoritmus helyességének bizonyítása (6.1. Propozíció). A logikai implikációt eldöntő 6.1. Algoritmus segítségével megadtam a logikai implikáció axiomatizációját Armstrong típusú axiómákkal. Példák segítségével bemutattam a reguláris nyelvekre alkalmazott funkcionális függőség (RFD) alkalmazását az XML sémanyelveire. Összehasonlítottam az RFD épségi megszorításokat kifejező erejét más XML funkcionális függőség (XFD) modellek kifejező erejével. Az RFD modellt az XML teljes leírására alkalmas kiterjesztett környezetfüggetlen nyelvű modellben is elhelyeztem, összetett értékű adatok bevezetésével kerülve el a fa-szerkezetű modell felhasználását. Az RFD modell a teljes általánosságot megszorító feltételeket (megkötés) használ, azért, hogy az üres adatok, végtelen ábécék, rendezetlen halmazok összehasonlítása, számlálók kezelése ne váljon szükségessé. Ezekkel a megkötésekkel a logikai implikáció kezelése is egyszerűbbé vált. Külön témakörként foglalkoztam a megkötések feloldásával. Sikerült megmutatni, hogy a megkötések elhagyásával is érvényesek, bár természetesen szűkített formában, az eredeti modell eredményei, a tetszőleges számú diszjunkciót tartalmazó reguláris kifejezésre épített RFD például, az XFD modelltől eltérően, végesen axiomatizálható, ha az üres jelsorozatot érvényes jelsorozatnak tekintjük.

Summary

The central idea of this PhD thesis is a new model (a new point of view) for the dependencies (integrity constraints) of XML databases. The traditional model for XML dependencies is the XML tree, that is, a "two-dimensional" one. Our new concept is to restrict the tree to one dimension, to a "flat" regular expression. We use this model to describe a database constructed from sentences of a regular language, we define and discuss dependencies on this database. Dependencies on relational databases can be specified as first order logic formulas. Representing regular languages as extended relations, we can generalize the relational databases to the databases of extended relations. We can define functional dependencies on this database admitting the states of the accepting automaton for a regular language as attributes (Definitions 4.4., 4.5., 4.6.). Viewing the graph for the accepting automaton of a regular language, the "colors" on the edges of an accepting traversal represent the sentences of the language, the vertices build up the sentences of an other, also regular (the dual) language. The regular functional dependency is principially a relation between two subgraphs of the accepting automaton (Definition 6.3.). Based upon the coloring of this graph the (Chase-like) 6.1. Algorithm can be used for deciding the logical implication of RFDs in quadratic time. Two graphs - of the accepting automaton and its transitiv closure - represent a counter-example (left side same color, right side different ones) the algorithm applies the dependencies from the set and tries to repair the counter-example. Our main result is that Algorithm 6.1. decides the logical implication correctly (Proposition 6.1.). The logical implication of RFDs could be finitely axiomatized by a set of Armstrong-type axioms using the 6.1. decision algorithm. We have showed by examples that the RFD complies well with XML schema languages. We have compared the expressiveness of RFD with a couple of XFD (XML functional dependency) models. We have investigated the RFD model in the scope of extended context free languages, which are fully capable to model XML, using complex values instead of tree structures. The standard model of RFD avoids comparing empty symbol strings, excludes infinite symbol sets, it allows only sorted lists for comparison, and, RFD does not use counters. The extended models are free from one or more of these restrictions. We could show that the results for the standard model could be transferred (appropriately adjusted) to the extended models. For instance, the RFDs based upon a regular expression containing any number of disjunctions can be finitely axiomatized (when we allow empty strings), this is not generally true for XFD models.